PATENT
Attorney Docket No.: 16869N-104800US
Client Ref. No.: NT1254US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

IKUYA YAGISAWA et al.

Application No.: 10/775,702

Filed: February 9, 2004

For: ARRAY-TYPE DISK APPARATUS PREVENTING DATA LOST WITH TWO DISK DRIVES FAILURE IN THE SAME RAID GROUP, THE PREVENTING PROGRAMMING AND SAID METHOD

Customer No.: 20350

Examiner: Unassigned

Technology Center/Art Unit: 2655

Confirmation No.: 9171

PETITION TO MAKE SPECIAL FOR NEW APPLICATION UNDER M.P.E.P. § 708.02, VIII & 37 C.F.R. § 1.102(d)

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is a petition to make special the above-identified application under MPEP § 708.02, VIII & 37 C.F.R. § 1.102(d). The application has not received any examination by an Examiner.

(a) The Commissioner is authorized to charge the petition fee of $130 under 37 C.F.R. § 1.17(i) and any other fees associated with this paper to Deposit Account 20-1430.

04/12/2005 AWONDAF1 00000100 201430 10775702
01 FC:1464 130.00 DA

04/12/2005 AWONDAF1 00000101 201430 10775702
01 FC:1464 130.00 DA

(b) All the claims are believed to be directed to a single invention. If the Office determines that all the claims presented are not obviously directed to a single invention, then Applicants will make an election without traverse as a prerequisite to the grant of special status.

(c) Pre-examination searches were made of U.S. issued patents, including a classification search and a key word search. The classification search was conducted on or around August 16, 2004 covering Class 714 (subclasses 6, 7, 42, 52, 54, and 800) and Class 711 (subclasses 113, 114, and 162), by a professional search firm, Lacasse & Associates, LLC. The key word search was performed on the USPTO full-text database including published U.S. patent applications. The inventors further provided three references considered most closely related to the subject matter of the present application (see references #7-9 below), which were cited in the Information Disclosure Statement filed with the application on February 9, 2004.

(d) The following references, copies of which are attached herewith, are deemed most closely related to the subject matter encompassed by the claims:

(1) U.S. Patent No. 6,070,249;

(2) U.S. Patent No. 6,243,827 B1;

(3) U.S. Patent No. 6,442,711 B1;

(4) U.S. Patent No. 6,647,514 B1;

(5) U.S. Patent Publication No. 2003/0056142 A1;

(6) U.S. Patent Publication No. 2003/0188101 A1;

(7) U.S. Patent No. 5,611,069;

(8) Japanese Patent Publication No. JP 08-147112; and

(9) David A. Patterson et al., "A Case for Redundant Arrays of Inexpensive Disks (RAID)," Computer Science Division, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, 1988.

(e) Set forth below is a detailed discussion of references which points out with particularity how the claimed subject matter is distinguishable over the references.

A. Claimed Embodiments of the Present Invention

The claimed embodiments relate to a disk drive which is an external memory device for a computer, and, more particularly, to a technique for preventing a plurality of disk drives in an array-type disk apparatus constituting a disk array from failing simultaneously and a technique for improving the host I/O response and improving the reliability at the time of data shifting among disk drives constituting a disk array group having a redundancy.

Independent claim 22 recites a storage system comprising a plurality of disks including first type disks configuring a RAID group and at least one second type disk, wherein each of the first type disks stores one of data received from a computer coupled to the storage system or parity data used for recovering the data received from the computer, and wherein the at least one second type disk is used as a spare disk for storing copy data of data stored in one of the first type disks; and a control section configured to hold an error status of each of the first type disks, start to mirror data between one of the first type disks and the at least one second type disk when the error status of the one of the first type disks matches a predetermined first criterion. After starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one second type disk, according to the error status of the one of the first type disks and the another one of the first type disks.

In this claimed embodiment, the first type disk to be configured to be mirroring pair with the second type disk is switched according to the error status of each first type disk. As a disk drive to be mirrored is dynamically switched, this operation is called "dynamic mirroring operation."

One of the benefits that may be derived is that it provides a highly reliable array-type disk apparatus which copies data to a spare disk drive for a possible failure and reduces the probability of occurrence of a 2 disk drives failure without involving a cost increase for spare disk drives.

B. Discussion of the References

1. U.S. Patent No. 6,070,249

This reference discloses a split party spare disk achieving method in raid subsystem. Discussed is a split parity disk achieving method for improving the defect endurance and performance of a RAID subsystem which distributively stores data in a disk array. Method may consist of constructing the disk array with at least two data disk drives for storing data, a spare disk drive used when a disk drive fails and a parity disk drive for storing parity data; and splitting the parity data of the parity disk drive and storing the split data in the parity disk drive and the spare disk drive. See column 3, line 61 to column 4, line 4.

The reference does not teach that after starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one second type disk, according to the error status of the one of the first type disks and the another one of the first type disks, as recited in independent claim 22. There is no disclosure of the "dynamic mirroring operation," in which a disk drive to be mirrored with the spare disk is switched.

2. U.S. Patent No. 6,243,827 B1

This reference relates to a multiple-channel failure detection in raid systems. Discussed are the use of software and a small portion of each disk in an array to write a bad area table on each disk. The embodiments may facilitate recovery of a RAID storage system from simultaneous failure of two or more disks. See column 4, lines 34-39. This reference does not appear to specifically utilize spare disks for failure protection, although it does address the situation of failure recovery from multiple disk failures.

The reference does not teach that after starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one

second type disk, according to the error status of the one of the first type disks and the another one of the first type disks, as recited in independent claim 22. There is no disclosure of the "dynamic mirroring operation," in which a disk drive to be mirrored with the spare disk is switched.

3. U.S. Patent No. 6,442,711 B1

This reference discloses a system and a method for avoiding storage failures in a storage array system. Discussed is a method for executing preventive maintenance of the conventional storage array system. A storage array system comprises a plurality of data storage devices for storing data, a spare storage device for replacing one of the plurality of data storage devices, and a control unit for controlling input and output operations. The control unit may include means for judging a necessity to execute preventive maintenance of each of the plurality of data storage devices by looking at the error rate. See column 2, lines 6-9, 22-27, and 32-35.

The disk array system includes data disks, parity disk, and spare disk. See Fig. 2. In this disk array system, the necessity to execute maintenance is judged by the error rate of each disk. If the result of the judgment is in need, data of the disk is copied to the spare disk. See Fig. 7. There is, however, no disclosure of switching the copy-originated disk (i.e., the disk to be mirrored with the spare disk) configured for copying the data to the spare disk by considering the error rate of the other disks.

The reference does not teach that after starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one second type disk, according to the error status of the one of the first type disks and the another one of the first type disks, as recited in independent claim 22. There is no disclosure of the "dynamic mirroring operation," in which a disk drive to be mirrored with the spare disk is switched.

4. U.S. Patent No. 6,647,514 B1

This reference discloses host I/O performance and availability of a storage array during rebuild by prioritizing I/O request. Discussed are rebuild I/O requests which may be given priority over host I/O requests when the storage array is close to permanently losing data (for example, failure of one more particular disk in the storage array would result in data loss). Due to different RAID levels, failures of different disks can result in different RAID levels being rebuilt. Examples of rebuilding data in an array include migrating to other disks and/or RAID levels, or writing data to a spare disk. See column 3, lines 47-51; and column 6, lines 11-13 and 17-23.

The reference does not teach that after starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one second type disk, according to the error status of the one of the first type disks and the another one of the first type disks, as recited in independent claim 22. There is no disclosure of the "dynamic mirroring operation," in which a disk drive to be mirrored with the spare disk is switched.

5. U.S. Patent Publication No. 2003/0056142 A1

This reference relates to a method and a system for leveraging spares in a data storage system including a plurality of disk drives. Disclosed is a method and system for leveraging spare disks for data redundancy in response to failure in a data storage system. The data storage system may be grouped into a plurality of arrays having data redundancy. The plurality of arrays may be arranged to maximize the number of arrays that are mirrored pairs of disk drives. In another embodiment, the plurality of arrays may be arranged in an optimum combination of arrays of mirrored pairs of disk drives. For every failure of one of said plurality of arrays due to a filed disk drive, a new array having data redundancy in a RAID configuration is created in the plurality of arrays. See paragraphs [0022]-[0025].

The reference does not teach that after starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to

stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one second type disk, according to the error status of the one of the first type disks and the another one of the first type disks, as recited in independent claim 22. There is no disclosure of the "dynamic mirroring operation," in which a disk drive to be mirrored with the spare disk is switched.

6. U.S. Patent Publication No. 2003/0188101 A1

This reference discloses partial mirroring during expansion thereby eliminating the need to track the progress of stripes updated during expansion. Discussed is a method of mirroring data form a write request to a spare unit corresponding to a stripe unit in a spare disk being rebuilt during expansion. The disk array may already be configured to enter a compaction state upon failure of replaced or spare disk during the expansion process since the spare units contain a copy of the valid data stored in the corresponding stripe units in the replaced or spare disk. See paragraphs [0014] and [0023].

The reference does not teach that after starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one second type disk, according to the error status of the one of the first type disks and the another one of the first type disks, as recited in independent claim 22. There is no disclosure of the "dynamic mirroring operation," in which a disk drive to be mirrored with the spare disk is switched.

7. U.S. Patent No. 5,611,069

This reference discloses a disk array apparatus which predicts errors using mirror disks that can be accessed in parallel. A mirror disk unit 36-1 in which two disk units are provided as one set is used as a component element of the disk array. The two disk units of the mirror disk unit 36-1 are allocated to the disk unit for the present use 32-1 and the disk unit for spare 32-2. Data is written into both the presently used disk unit 32-1 and the spare disk unit 32-2. Data is read out from the present use disk unit 32-1. See Fig. 1 and column 8, lines 5-24. The occurrence of a fault of the disk unit is judged and the allocation is switched

from the present use disk unit to the spare disk unit. In an idle state, a simulation to check the disk array is executed and fault information is collected. See column 12, lines 5-53. The present use disk unit is not constructed as a mirror disk and when the fault is judged in the present use disk unit, the data is copied to the spare disk unit, thereby dynamically realizing a mirror disk construction. See Figure 18 and column 16, lines 18-21 and 40-47.

The reference does not teach that after starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one second type disk, according to the error status of the one of the first type disks and the another one of the first type disks, as recited in independent claim 22. There is no disclosure of the "dynamic mirroring operation," in which a disk drive to be mirrored with the spare disk is switched.

8. Japanese Patent Publication No. JP 08-147112

This reference discloses a technique to efficiently perform the error recovery work by automatically performing the recovery processing. If the frequency in error occurrence of one of disk devices 50-57 for data storage and a disk device 58 for redundant information storage in a disk array 5 exceeds a prescribed value, data of the disk device where error occurs is restored into an auxiliary disk device 59 by a first data restoration part 46; and when the restoration operation of this part 46 is completed, a reinitializing part 47 initializes (formats) the medium of the disk device where the error occurs. After initialization of the reinitializing part 47 is completed, a medium check part 48 checks the medium of the disk device where the error occurs. A second data restoration part 49 restores data of the auxiliary disk device 59 into an error disk device when it is discriminated by the medium check part 48 that the medium is normal.

As discussed in the present application at page 2, line 27 to page 3, line 27, the reference discloses a technique which copies data of a disk drive to its spare disk drive and restores the data in the spare disk drive in case where the number of errors occurred in that disk drive exceeds a specified value. Further, the conventional array-type disk apparatus has an operational flow such that when a data read failure occurs frequently in a disk drive from

which data is shifted (hereinafter called "data-shifting disk drive") at the time of shifting data to the spare disk drive of the disk drive due to preventive maintenance or so, data read from the data-shifting disk drive is attempted and after a data read failure is detected, the data in the data-shifting disk drive is restored by the disk drive that has redundancy using the data restoring function of the array-type disk apparatus. It is therefore expected that the prior art drive suffers a slower response to the data read request from the host computer. To avoid the response drop, it is typical to perform the process of coping with the data read request from the host computer using only the system which isolates the data-shifting disk drive from the array-type disk apparatus when a data read error has occurred frequency in the data-shifting disk drive and restores the data in the data-shifting disk drive by means of the redundant disk drive by using the data restoring function of the array-type disk apparatus.

The reference does not teach that after starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one second type disk, according to the error status of the one of the first type disks and the another one of the first type disks, as recited in independent claim 22. There is no disclosure of the "dynamic mirroring operation," in which a disk drive to be mirrored with the spare disk is switched.

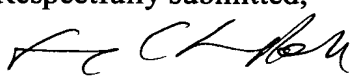
9. David A. Patterson et al., "A Case for Redundant Arrays of Inexpensive Disks (RAID)," Computer Science Division, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, 1988

This reference discloses redundant arrays of inexpensive disks (RAID). As discussed in the present application at page 1, line 16 to page 2, line 5, the reference discloses an array-type disk apparatus known as a RAID (Redundant Arrays of Inexpensive Disks) and is a memory device which has a plurality of disk drives laid out in an array and a control section to control the disk drives. In the array-type disk apparatus, a read request (data read request) and a write request (data write request) are processed fast by the parallel operation of the disk drives and redundancy is added to data. Array-type disk apparatuses are classified into five levels according to the type of redundant data to be added and the structure.

The reference does not teach that after starting to mirror data between the one of the first type disks and the at least one second type disk, the control section is configured to stop mirroring data between the one of the first type disks and the at least one second type disk and start to mirror data between another one of the first type disks and the at least one second type disk, according to the error status of the one of the first type disks and the another one of the first type disks, as recited in independent claim 22. There is no disclosure of the "dynamic mirroring operation," in which a disk drive to be mirrored with the spare disk is switched.

(f) In view of this petition, the Examiner is respectfully requested to issue a first Office Action at an early date.

Respectfully submitted,



Chun-Pok Leung
Reg. No. 41,405

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 650-326-2400
Fax: 415-576-0300
Attachments
RL:rl
60422079 v1

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-147112
(43)Date of publication of application : 07.06.1996

(51)Int.CI. G06F 3/06
G06F 3/06
G06F 3/06
G11B 20/18

(21)Application number : 06-286189
(22)Date of filing : 21.11.1994

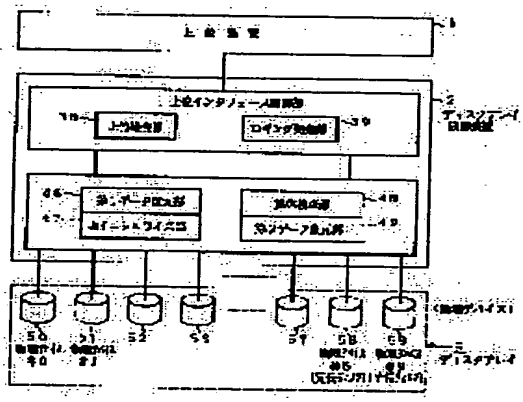
(71)Applicant : FUJITSU LTD
(72)Inventor : KONNO SHIGEO

(54) ERROR RECOVERY DEVICE FOR DISK ARRAY DEVICE

(57)Abstract:

PURPOSE: To efficiently perform the recovery work by automatically performing the recovery processing without requiring hands neither replacement of a disk device by medium initialization of the disk device where a fault occurs.

CONSTITUTION: If the frequency in error occurrence of one of disk devices 50 to 57 for data storage and a disk device 58 for redundant information storage in a disk array 5 exceeds a prescribed value, data of the disk device where error occurs is restored into an auxiliary disk device 59 by a first data restoration part 46; and when the restoration operation of this part 46 is completed, a re-initializing part 47 initializes (formats) the medium of the disk device where the error occurs. After initialization of the re-initializing part 47 is completed, a medium check part 48 checks the medium of the disk device where the error occurs. A second data restoration part 49 restores data of the auxiliary disk device 59 into an error disk device when it is discriminated by the medium check part 48 that the medium is normal.



LEGAL STATUS

[Date of request for examination] 19.10.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平8－147112

(43)公開日　平成８年(1996)６月７日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　3/06	５４０			
	３０４　Ｐ			
	３０６　Ｂ			
Ｇ１１Ｂ　20/18	５７０　Ｚ	8940－5D		

審査請求　未請求　請求項の数４　ＯＬ　（全 11 頁）

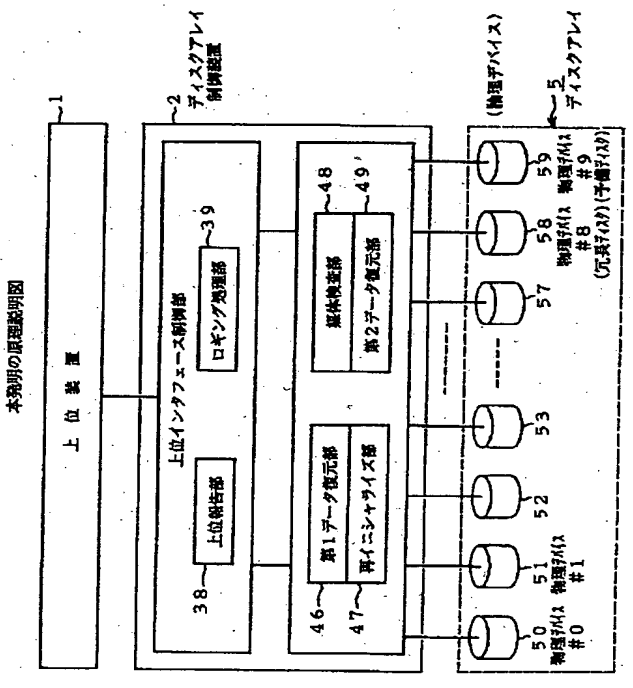
(21)出願番号　特願平6－286189

(22)出願日　平成６年(1994)11月21日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地
(72)発明者　金野　茂生
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(74)代理人　弁理士　竹内　進　（外１名）

(54)【発明の名称】　ディスクアレイ装置のエラー回復装置

(57)【要約】
【目的】障害発生ディスク装置の媒体イニシャライズによる回復処理を人手やディスク装置の交換を必要とすることなく自動的に行って復旧作業を効率化する。
【構成】ディスクアレイ５のデータ記憶用ディスク装置５０～５７及び冗長情報記憶用ディスク装置５８のいずれかのエラー発生回数が規定値を越えた場合に、第１データ復元部４６により、エラー発生ディスク装置のデータを予備ディスク装置５９に復元する。データ復元部４６による復元動作が完了したら、再イニシャライズ部４７が、エラーディスク装置の媒体をイニシャライズ（フォーマッティング）する。更に媒体検査部４８で、再イニシャライズ部４７によるイニシャライズが完了した後に、エラーディスク装置の媒体の検査を行う。第２データ復元部４９は、媒体検査部４８により媒体正常が判定された場合に、予備ディスク装置５９のデータをエラーディスク装置に復元する。



【特許請求の範囲】

【請求項１】データ記憶用と冗長情報記憶用の複数のディスク装置を備えたディスクアレイを接続し、上位装置からのアクセスに対して前記複数の磁気ディスク装置を並列アクセスするディスクアレイ制御装置を有し、更に、前記ディスクアレイは少なくとも１台の予備ディスク装置を備えたディスクアレイ装置に於いて、

前記ディスクアレイ制御装置に、

前記ディスクアレイのデータ記憶用及び冗長情報記憶用の複数のディスク装置のいずれかのエラー発生回数が規定値を越えた場合に、エラー発生ディスク装置のデータを前記予備ディスク装置に復元する第１データ復元部と、

前記データ復元部による復元動作が完了した後に、前記エラーディスク装置の媒体をイニシャライズする再イニシャライズ部と、

前記再イニシャライズ部によるイニシャライズが完了した後に、前記エラーディスク装置の媒体の検査を行う媒体検査部と、

前記媒体検査部により媒体正常が判定された場合に、前記予備ディスク装置のデータをエラーディスク装置に復元する第２データ復元部と、を設けたことを特徴とするディスクアレイ装置のエラー回復装置。

【請求項２】請求項１記載のディスクアレイ装置のエラー回復装置に於いて、更に、前記第１データ復元部によるデータ復元の開始と終了、前記再イニシャライズ部による再イニシャライズの開始、前記媒体検査部による媒体正常判定に基づく再イニシャライズの終了、前記第２データ復元部によるデータ復元の開始と終了の各々を、上位装置に通知する上位報告部を設けたことを特徴とするディスクアレイ装置のエラー回復装置。

【請求項３】請求項２記載のディスクアレイ装置のエラー回復装置に於いて、前記上位報告部は、前記第１データ復元部によるデータ復元、前記再イニシャライズ部による再イニシャライズ、及び前記第２データ復元部によるデータ復元の各々について、上位装置への完了報告からの経過時間を監視し、一定時間を越えても前記上位装置又はオペレータからの指示がない場合は、強制的に次の処理に移行させることを特徴とするディスクアレイ装置のエラー回復装置。

【請求項４】請求項１記載のディスクアレイ装置のエラー回復装置に於いて、更に、前記第１データ復元部によるデータ復元、前記再イニシャライズ部による再イニシャライズ、及び前記第２データ復元部によるデータ復元の各々の報告内容を、不揮発性記憶部に記憶保持するロギング処理部を設けたことを特徴とするディスクアレイ装置のエラー回復装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、データ記憶用又はパリティ記憶用のディスク装置の障害発生時に予備ディスク装置へデータ復元して対応するディスクアレイ装置のエラー回復装置に関し、特に、予備ディスク装置に切り替えた後にエラーディスク装置の回復動作を試みるようにしたディスクアレイ装置のエラー回復装置に関する。

【０００２】

【従来の技術】高速化、高性能化が進む近年のコンピュータシステムにおいて、半導体テクノロジの進歩を背景とした中央処理装置の性能向上は目覚ましいものがあり、このため、外部に接続される外部記憶装置に対しても同様な高性能化が要求されている。この要求に対して、機械的動作を伴う磁気ディスク装置の高速化に限界があるため、複数の磁気ディスク装置でディスクアレイを構成してディスクアレイ制御装置に並列接続し、複数の磁気ディスク装置を並列アクセスしてリード、ライト動作を行うディスクアレイ装置が提供されている。

【０００３】このようなディスクアレイ装置では、運用されているディスク装置に対し、予備ディスク装置を設け、運用ディスク装置の障害時に予備ディスク装置に切り替えて運用する。図６は従来のディスクアレイ装置である。ディスクアレイ制御装置２は、上位装置１と接続される上位装置インタフェース制御部３と、ディスクアレイ５の複数の磁気ディスク装置５０～５９と接続されるデバイス制御部４により構成される。ディスクアレイ５は、データ記憶用ディスク装置５０～５７と冗長情報記憶用ディスク装置（以下「冗長ディスク装置」という）５７を有し、さらに予備ディスク装置５９を設けている。

【０００４】ディスクアレイ制御装置２は、上位装置１からのデータ転送要求に対して、デバイス制御部４を経由して磁気ディスク装置５０～５８を並列にアクセスし、リード処理またはライト処理を同時に行う。即ち、複数のデータ記憶用ディスク装置５０～５７にデータが書き込まれる際に、冗長ディスク装置５８に対してパリティデータ等を生成して書込みを行う。パリティは、データの読出し時に複数のデータ記憶用ディスク内のある一台の磁気ディスク装置において何らかの障害が発生した場合においても、他の正常なディスク装置のデータと冗長ディスク装置のパリティデータからのデータ復元を可能としている。

【０００５】また、ディスク装置５０～５８の内のある一台のディスク装置において連続して障害が発生した場合、デバイス制御部４の指示により障害を起こしたエラーディスク装置を論理ディスクの割当てから切り離して予備ディスク装置に割り当て、新たに割り当てた予備ディスク装置にエラーディスク装置の全データを復元させている。

【０００６】予備ディスク装置に対するデータ復元処理は、オペレータによる指示も可能であるが、通常はディスクアレイ制御装置２にてエラーの発生状況を監視し、

エラーの発生がある一定値を越えた場合に自動的にデータ復元を開始させている。

【０００７】

【発明が解決しようとする課題】ところで、ディスク装置に発生する障害としては、ディスク装置を構成する部品等の劣化や不良等によるところが多いが、部品を構成する材料等の特性やトラックずれ等により訂正不可能なデータチェックが発生することがある。一般にこれらのデータチェック障害は、媒体のイニシャライズ処理（フォーマット処理）により復旧することが可能である。

【０００８】しかし、従来装置にあっては、障害を起こしたディスク装置は新品と交換することを前提としており、媒体のイニシャライズ処理で復旧可能な障害であっても、必ずシステム筐体から障害ディスク装置を外し、別の試験装置等にセットしてイニシャライズしてみなければならず、ディスク装置の交換や、イニシャライズのための人手による作業を必要としていたため、ディスクアレイ装置の復旧作業に時間がかかるという問題があった。

【０００９】本発明は、障害を起こしたディスク装置の媒体イニシャライズによる回復処理を人手やディスク装置の交換を必要とすることなく自動的に行って障害発生に対する復旧作業を効率化して短時間で処理できるようにしたディスクアレイ装置のエラー回復装置を提供することを目的とする。

【００１０】

【課題を解決するための手段】図１は本発明の原理説明図である。まず本発明は、データ記憶用ディスク装置５０～５７と冗長情報記憶用ディスク装置５８を備えたディスクアレイ５を接続し、上位装置１からのアクセスに対して複数の磁気ディスク装置５０～５８を並列アクセスするディスクアレイ制御装置２を有し、更に、ディスクアレイ５は少なくとも１台の予備ディスク装置５９を備えたディスクアレイ装置を対象とする。

【００１１】このようなディスクアレイ装置のエラー回復装置として本発明にあっては、ディスクアレイ制御装置２に、第１データ復元部４６、再イニシャライズ部４７、媒体検査部４８及び第２データ復元部４９を設ける。第１データ復元部４６は、ディスクアレイ５のデータ記憶用ディスク装置５０～５７及び冗長情報記憶用ディスク装置５８のいずれかのエラー発生回数が規定値を越えた場合に、エラー発生ディスク装置のデータを予備ディスク装置５９に復元する。再イニシャライズ部４７は、第１データ復元部４６による復元動作が完了した後に、エラーディスク装置の媒体をイニシャライズ（フォーマッティング）する。媒体検査部４８は、再イニシャライズ部４７によるイニシャライズが完了した後に、エラーディスク装置の媒体の検査を行う。第２データ復元部４９は、媒体検査部４８により媒体正常が判定された場合に、予備ディスク装置５９のデータをエラーディスク装置に復元する。

【００１２】更に、ディスクアレイ制御装置２に上位報告部３８を設け、第１データ復元部４６によるデータ復元の開始と終了、再イニシャライズ部４７による再イニシャライズの開始、媒体検査部４８による媒体正常判定に基づく再イニシャライズの終了、第２データ復元部４９によるデータ復元の開始と終了の各々を、上位装置１に通知する。

【００１３】更に、上位報告部３８は、第１データ復元部４６によるデータ復元、再イニシャライズ部４７による再イニシャライズ、及び第２データ復元部４９によるデータ復元の各々について、上位装置１への完了報告からの経過時間を監視し、一定時間を越えても上位装置１又はオペレータからの指示がない場合は、強制的に次の処理に移行させる。

【００１４】更に、ディスクアレイ制御装置２にロギング処理部３９を設け、第１データ復元部４６によるデータ復元、再イニシャライズ部４７による再イニシャライズ、及び第２データ復元部４９によるデータ復元の各々の報告内容を、不揮発性記憶部に記憶保持する。

【００１５】

【作用】このような本発明によるディスクアレイ装置のエラー回復装置によれば次の作用が得られる。ディスクアレイのデータ及び冗長記録用のディスク装置のいずれかでエラーが多発してエラー発生回数が規定値を越えたときに、自動的に予備ディスク装置へのデータ復元動作を開始する。このとき上位装置に対してデータ復元の開始が通知される。予備ディスク装置へのデータ復元動作が完了すると、上位装置にその旨を報告し、完了報告に対する指示を時間監視により待つ。

【００１６】上位装置またはオペレータからの指示があるか、或いは監視時間がオーバフローすると、エラーディスク装置の再イニシャライズを実施する。再イニシャライズが済むと、次にイニシャライズが済んだ媒体を検査する検査処理（診断処理）を行う。媒体診断が正常であれば、この時点で上位装置に再イニシャライズの完了を報告し、完了報告に対する指示を時間監視により待つ。

【００１７】上位装置またはオペレータからの指示があるか、或いは監視時間がオーバフローすると、予備ディスク装置から再イニシャライズによりエラーの回復したディスク装置にデータを復元し、上位装置に対してデータ復元完了を通知する。これにより媒体のイニシャライズで回復可能なディスク装置の故障を、ディスク装置を交換したり人手を必要とすることなく、正常なディスク装置に回復させることができる。

【００１８】また上位装置からディスクアレイ制御装置が切り離されても、ロギング処理部によりディスクアレイ制御装置の不揮発性記憶部に障害発生に対する復旧状況及び結果が格納され、ロギング情報として上位装置に

提供することができる。

【００１９】

【実施例】図２は、本発明の一実施例を示したブロック図である。図２において、本発明のディスクアレイ装置は、上位装置としてのホストコンピュータ１に接続されたディスクアレイ制御装置２と、論理デバイスとして複数のディスク装置５０～５９を並列接続したディスクアレイ５から構成される。ディスクアレイ５は、この実施例にあっては、データを記憶するための８台の記憶用ディスク装置５０～５７、１台のパリティ情報を記憶する冗長ディスク装置５８、および１台の予備ディスク装置５９で構成される。

【００２０】ディスクアレイ装置２は、ホストコンピュータ１と接続される上位インタフェース制御部３と、ディスクアレイ５と接続されるデバイス制御部４で構成される。上位インタフェース制御部３には、インタフェース制御部３１、ＭＰＵ３２、データ転送制御部３３、フラグレジスタ３５、カウンタ３４、不揮発記憶部３６が設けられる。

【００２１】ＭＰＵ３２は、マイクロプログラム３７によりホストコンピュータ１からのデータ転送要求に対する各種の処理を行い、その処理機能の中に、デバイス制御部４によるディスクアレイ５の状態、特にエラー回復処理に伴う各種の状態や結果をホストコンピュータ１に報告するための上位報告部３８と、上位報告部３８で報告するエラー回復の状況や結果を不揮発記憶部３６にロギング情報として記憶保持するロギング処理部３９の機能を設けている。更にＭＰＵ３２には、オペレータ制御部６が接続され、エラー回復などの各種のメンテナンスに必要な情報をオペレータがオペレータ制御部６よりＭＰＵ３２に指示可能としている。

【００２２】デバイス制御部４には、ディスクアレイ制御部４１、ＭＰＵ４２、データ転送制御部４３、データチェックカウンタ４４が設けられる。ＭＰＵ４２は、マイクロプログラム４５を実行し、上位インタフェース制御部３のＭＰＵ３２によるホストコンピュータ１からのデータ転送要求に伴うディスクアレイ５に対するリード動作またはライト動作、更に本発明のエラー回復のたの処理動作を行う。

【００２３】このエラー回復のため、マイクロプログラム４５には、第１データ復元部４６、再イニシャライズ部４７、媒体検査部４８および第２データ復元部４９の各機能が設けられている。データチェックカウンタ４４は、ディスクアレイ５に設けたディスク装置５０～５９ごとにカウンタ領域をもっており、ホストコンピュータ１からのデータ転送要求に伴うディスクアレイのアクセス時のリード動作で得られた読出データについて、ＥＣＣにより訂正不可能なエラーを検出したときに障害発生と判断して、エラーを起こしたディスク装置に対応するデータチェックカウンタ４４の値を１つインクリメントする。

【００２４】第１データ復元部４６は、データチェックカウンタ４４の計数値を監視しており、エラー発生回数が予め定めた規定値に達すると、エラー回数が規定値に達したディスク装置をエラーディスク装置と判定し、エラー回復処理の対象に指定し、エラーディスク装置のデータを予備ディスク装置５９に復元させるためのデータ復元処理を実行する。

【００２５】予備ディスク装置５９に対するデータ復元は、エラーディスク装置を除く正常な記憶用ディスク装置と冗長ディスク装置５８の各データを使用して生成することができる。再イニシャライズ部４７は、第１データ復元部４６で予備ディスク装置５９に対するエラーディスク装置のデータ復元が正常終了した場合のホストコンピュータ１またはオペレータ制御部６からの指示、あるいはいずれの指示もない場合は、上位インタフェース制御部３に設けたカウンタ３４による時間監視でオーバフローした際に起動し、エラーディスク装置の媒体の再イニシャライズ、即ち初期化処理としてのフォーマッティングを実行させる。

【００２６】媒体検査部４８は、再イニシャライズ部４７によるエラーディスク装置の媒体のイニシャライズが終了した時点で起動し、イニシャライズが済んだ媒体のデータ面に所定のダミーデータを全面に書き込み、続いて全面のリードを行って、正常にリードできたか否かの媒体検査を行う。媒体検査部４８による検査が正常に終了すれば、これで再イニシャライズの完了となる。再イニシャライズの完了は、ホストコンピュータ１およびオペレータ制御部９に報告される。

【００２７】第２データ復元部４９は、再イニシャライズ完了後にホストコンピュータ１またはオペレータ制御部６からの指示、あるいは上位インタフェース制御部３に設けたカウンタ３４による時間監視がオーバフローした際に起動し、再イニシャライズが済んで正常に動作可能な、エラーを起こしたディスク装置に対し、予備のディスク装置５９のデータを復元する。この場合、予備のディスク装置５９は正常に動作していることから、予備のディスク装置５９のデータをエラー回復が済んだディスク装置にコピーすることになる。

【００２８】更に、上位インタフェース制御部３のＭＰＵ３２の機能として設けた上位報告部３８は、デバイス制御部４のＭＰＵ４２による第１データ復元部４６、再イニシャライズ部４７、媒体検査部４８および第２データ復元部４９によるエラー回復処理の開始と終了およびその結果をホストコンピュータ１に報告する。なお、再イニシャライズについては、その開始は再イニシャライズ部４７による動作開始を報告し、再イニシャライズの終了は媒体検査部４８による正常終了で再イニシャライズ完了を報告することになる。

【００２９】上位報告部３８は、ホストコンピュータ１

に加えて、必要に応じてオペレータ制御部６にエラー回復処理の状況および結果を報告することができる。例えば、保守要員がディスクアレイ制御装置２についている場合には、オペレータ制御部６に状況を報告して操作パネルなどに所定のコード番号による状態表示や結果表示を行い、オペレータのエラー回復に対する指示を待つことができる。

【００３０】更に上位報告部３８は、ホストコンピュータ１に対するエラー回復のための各種の動作の開始報告を行った際に、カウンタ３４を起動して時間監視を行い、カウンタ３４の計数値が一定時間後にオーバフローすると、ホストコンピュータ１またはオペレータ制御部６からの指示を待つことなく、ＭＰＵ４２に対し次のエラー回復のための処理への移行を指示する。

【００３１】上位報告部３８によるホストコンピュータ１への報告処理は、フラグレジスタ３５の状態に応じて行われる。フラグレジスタ３５が１にセットされている場合、上位報告部３８は割込処理によりホストコンピュータ１に対する報告を行う。これに対しフラグレジスタ３５が０にリセットされている場合には、ホストコンピュータ１からのアクセスに対する応答ステータスとして上位装置への報告を行うことになる。

【００３２】即ち、ディスクアレイ制御装置２がホストコンピュータ１から切り離されている状態では、フラグレジスタ３５は１にセットされており、この状態では割込みによりホストコンピュータ１への報告が行われる。一方、ホストコンピュータ１とディスクアレイ制御装置２が結合されてデータ転送中にあっては、例えば転送終了時のステータス情報に含めて上位装置への報告を行うようになる。

【００３３】図３は、図２のディスクアレイ制御装置２によるデータ転送処理の概略である。まずステップＳ１で、上位インタフェース制御部３のＭＰＵ３２がホストコンピュータ１からのデータ転送による入出力要求の有無をチェックしている。入出力要求があると、ステップＳ２に進み、デバイス制御部４のＭＰＵ４２に対しリードコマンドまたはライトコマンドを発行し、ディスクアレイ制御部４１を介して、ディスクアレイ５の記憶用ディスク装置５０～５７、更に冗長ディスク装置５８の並列アクセスによるステップＳ２のリード動作またはライト動作を行う。

【００３４】例えば、ホストコンピュータ１からのライトデータの転送要求に対しては、チャネルインタフェース制御部３１、データ転送制御部３３、データ転送制御部４３、ディスクアレイ制御部４１を経由して、記憶用ディスク装置５０～５７に対するデータ書込みおよび冗長ディスク装置５８に対するパリティデータの書込みが行われる。

【００３５】また、ホストコンピュータ１からのリードデータ転送要求に対しては、ディスクアレイ５の記憶用ディスク装置５０～５７よりデータの読出しを行い、ディスクアレイ制御部４１、データ転送制御部４３、データ転送制御部３３、チャネルインタフェース制御部３１を経由して、ホストコンピュータに要求データを転送する。

【００３６】次にステップＳ３で、ディスクアレイ５の運用中のディスク装置において、訂正不可能なエラーが発生したディスクがあるか否かチェックする。もし訂正不可能なエラーを発生したディスク装置があれば、ステップＳ４に進み、ＭＰＵ４２がデータチェックカウンタ４４の対応するカウンタエリアのエラー発生回数を１つインクリメントする。

【００３７】次にステップＳ５で、データチェックカウンタ４４の値の中に予め定めた規定値を越えるエラー発生回数のディスク装置があるか否かチェックする。もし規定値を越えるエラー発生回数のディスク装置があれば、そのディスク装置をエラーディスクと判定し、ステップＳ６のエラー処理に進む。図４および図５は、図３のステップＳ６の本発明によるエラー処理の詳細である。このエラー処理について、図２のディスクアレイ５に設けている記憶用ディスク装置５０のエラー発生回数が規定値に達してエラーディスクと判定された場合を例にとって説明する。

【００３８】ＭＰＵ４２において、記憶用ディスク装置５０のデータチェックカウンタ４４の値が規定値に達すると、エラーディスクと判定して、ＭＰＵ３２に障害通知報告を行う。この障害通知報告を受けたＭＰＵ３２は、ＭＰＵ４２に対し、図４のステップＳ１に示すように、エラーディスク装置５０のデータを予備ディスク装置５９に復元させるためのデータ復元処理の開始を指示する。

【００３９】同時にＭＰＵ３２は、上位報告部３８の機能によりホストコンピュータ１に対しデータ復元処理が開始されたことを、ステップＳ２のように報告する。このときＭＰＵ３２は、フラグレジスタ３５の状態をチェックし、フラグレジスタ３５が１にセットされていれば、割込みによりホストコンピュータ１にデータ復元処理の開始を報告し、一方、フラグが０にリセットされていれば、現在行われているホストコンピュータ１からのアクセス終了に伴うステータス情報に含めてデータ復元処理の開始を報告する。

【００４０】ＭＰＵ３２からのデータ復元開始の指示を受けたＭＰＵ４２は、第１データ復元部４６の機能により、ディスクアレイ制御部４１を介してエラーディスク装置５０のデータを予備ディスク装置５９に復元するための復元処理を開始させる。このデータ復元処理は、エラーディスク装置５０を除いた正常な記憶用ディスク装置５１～５７の各データと冗長ディスク装置５８のパリティデータに基づいて生成することができる。

【００４１】予備ディスク装置５９に対するエラーディ

スク装置５０の全てのデータが復元して正常終了がステップＳ３で判別されると、ステップＳ４に進み、ＭＰＵ４２はＭＰＵ３２にデータ復元の完了報告を行う。これを受けてＭＰＵ３２は、そのときのフラグレジスタ３５の状態を参照しながら、ホストコンピュータ１に対するデータ復元完了報告を行う。

【００４２】ＭＰＵ３２は、ホストコンピュータ１に対するデータ完了報告を終わると、ステップＳ５で、ホストコンピュータ１からの確認応答を待っており、確認応答が得られて初めて報告完了と判断し、次のステップＳ６に進む。このホストコンピュータ１からの確認応答待ちの間は、ステップＳ６でロギング処理部３９を起動し、不揮発記憶部３６に予備ディスク装置５９に対するデータ復元完了の状態を記録する内部ロギング処理を行う。

【００４３】ステップＳ５で、ホストコンピュータ１から正常に確認応答が得られて報告完了になると、ステップＳ６に進み、ＭＰＵ３２はカウンタ３４を起動して時間監視を開始する。カウンタ３４は、予め定めた所定時間を経過するとオーバフローして、監視時間が終了したことを表わす。カウンタ３４がオーバフローする監視時間以内に、ホストコンピュータ１またはオペレータ制御部６より再イニシャライズの指示があれば、次のステップＳ７の処理に進む。また再イニシャライズの指示がなくとも、カウンタ３４がオーバフローした時点でＭＰＵ４２に再イニシャライズを指示することになる。

【００４４】ＭＰＵ４２は、ＭＰＵ３２によるホストコンピュータ１またはオペレータ制御部６による指示に基づいた再イニシャライズ、あるいは指示がないときのカウンタ３４のオーバフローに基づく再イニシャライズの指示を受け、ステップＳ７で、エラーディスク装置５０の媒体の再イニシャライズを指示する。この指示を受けて、ディスクアレイ制御部４１を介してエラーディスク装置５０は、工場出荷時と同様に媒体のフォーマッティングを再度やり直すイニシャライズ動作を開始する。

【００４５】エラーディスク装置５０の再イニシャライズの正常終了がステップＳ８で判別されると、ＭＰＵ４２は、続いてエラーディスク装置５０に対し、ステップＳ９で、再イニシャライズが済んだ媒体のデータ領域全面にダミーデータを書き込んだ後に全面をリードして、リード結果をチェックする媒体検査処理の開始を指示する。

【００４６】続いてステップＳ１０で、エラーディスク装置５０における媒体検査処理の正常終了がＭＰＵ４２で判別されると、ＭＰＵ４２はＭＰＵ３２に再イニシャライズ処理の完了を報告する。これを受けてＭＰＵ３２は、そのときのフラグレジスタ３５の状態に応じてホストコンピュータ１に対し再イニシャライズ処理の完了報告をステップＳ１１のように行う。

【００４７】再イニシャライズ処理の完了報告に対し、次のステップＳ１２で、ホストコンピュータ１より確認応答があるか否か監視しており、その間に、ステップＳ２２で、再イニシャライズ処理の完了を不揮発記憶部３６に内部ロギング処理として記憶保持させる。ホストコンピュータ１より確認応答を受けてステップＳ１２で報告完了が判別されると、ステップＳ１３で、ＭＰＵ３２はカウンタ３４をリセットして再度スタートし、ホストコンピュータ１またはオペレータ制御部６からの指示を受けるための時間監視を開始する。カウンタ３４がオーバフローする前に指示があれば、図５のステップＳ１４に進む。指示がなくとも、ステップＳ２３で一定時間後にカウンタ３４がオーバフローすれば、図５のステップＳ１４に進む。

【００４８】図５のステップＳ１４にあっては、ホストコンピュータ１またはオペレータ制御部６からの指示あるいはこの指示がなくとも、カウンタ３４のオーバフローに基づき、再イニシャライズが正常終了したエラーを起こしたディスク装置５０に対する予備ディスク装置５９からのデータ復元指示をＭＰＵ４２に対し行い、データ復元処理が開始される。

【００４９】続いて、予備ディスク装置５９のデータのディスク装置５０に対するエラー回復の正常終了をステップＳ１５でＭＰＵ４２が判別すると、このデータ回復処理の正常終了をＭＰＵ３２に通知する。ＭＰＵ３２は、そのときのフラグレジスタ３５の状態に応じホストコンピュータ１に、エラーを起こしたディスク装置５０の復旧処理の完了報告をステップＳ１６のように行う。

【００５０】続いてステップＳ１７で、ホストコンピュータ１からの確認応答を待っており、その間にステップＳ２５で、不揮発記憶部３６に、エラーを起こしたディスク装置５０が回復してデータ復元が完了したことを記録する内部ロギング処理を行う。ホストコンピュータ１より復旧処理の完了報告に対する確認応答がステップＳ１７で判別されると、一連のエラー発生に伴う回復処理を終了し、図３のメインルーチンにリターンする。

【００５１】一方、ステップＳ３で、予備ディスク装置５９に対するエラーディスク装置５０のデータ復元が正常終了できなかった場合には、予備ディスク装置５９に障害があることから、ステップＳ１８のエラー処理に進む。この場合には、エラーディスク装置５０に加えて予備ディスク装置５９を交換し、必要なデータ復元処理を行う。

【００５２】またステップＳ８で再イニシャライズが正常終了しなかったり、ステップＳ１０で媒体検査処理が正常終了しなかった場合には、ステップＳ２１で、エラーディスク装置５０は再イニシャライズを行っても使用できない障害を起こしているものと判断し、エラーディスク装置５０の交換によるエラー処理を行う。更に、ステップＳ１５において、再イニシャライズ完了後のエラーを起こしたディスク装置への予備ディスク装置５９か

らのデータ復元が正常終了できなかった場合には、ステップＳ２４で、ディスク装置５０に再イニシャライズでは回復できない別の障害が発生したものと判断し、ディスク装置５０を交換するエラー処理を行うことになる。

【００５３】尚、上記の実施例は、磁気ディスク装置を用いたディスクアレイを例にとっているが、光ディスク装置、半導体メモリ装置など適宜の物理デバイスを用いたアレイ装置に適用できる。また、ディスクアレイ５に設けた記憶用ディスク装置の台数は、必要に応じて適宜に定めることができる。また、実施例のディスクアレイ５は１ランク構成を例にとっているが、並列構成を多段階に設けた複数ランク構成としてもよい。

【００５４】更に上記の実施例にあっては、ホストコンピュータ１に対し

①エラーディスク装置から予備ディスク装置へのデータ復元の開始

②エラーディスク装置から予備ディスク装置へのデータ復元の完了

③エラーディスク装置の再イニシャライズ処理の完了

④エラーディスク装置に対する予備ディスク装置からのデータ復元の完了

を報告しているが、少なくとも最初の①のデータ復元開始報告と最後の④の復旧処理の完了を報告できればよく、その間の報告は必要に応じて適宜に定めることができる。

【００５５】特に本発明にあっては、上位装置に報告を行って指示を待つが、指示がなくともカウンタのオーバフローによる時間監視で次のエラー回復の処理に自動的に移行できるため、基本的には上位装置への状況の報告を行う必要はない。但し、上位装置からディスクアレイ制御装置２側の状態が見えなくなるのを回避するため、少なくとも不揮発記憶部３６にエラー回復のロギング情報を記憶させる必要はある。

【００５６】

【発明の効果】以上説明してきたように本発明によれば、訂正不可能なデータチェックの発生により、ディスクアレイの中のディスク装置のデータ復元が開始されると、ディスクアレイ制御装置の内部処理により上位装置またはオペレータからの作業指示を必要とすることなく、自動的に、エラーを起こしたディスク装置を可能な限り使用可能状態に戻す再イニシャライズを含む復旧作業が行われ、オペレータ不在などにより障害の復旧が遅れることなく実施され、更に、人手による操作ミスを防ぐことができる。

【００５７】またエラー発生ディスクについては、自動的に再イニシャライズと再イニシャライズ完了後の全面リード動作による媒体検査が行われ、正常終了でエラーは回復したものとして予備のディスク装置のデータを復元して、元の運用状態に自動的に戻るようになり、媒体のイニシャライズで回復するようなデータチェックの発生に対し効率良くエラー回復処理を行うことができる。

【図面の簡単な説明】

【図１】本発明の原理説明図

【図２】本発明の一実施例を示したブロック図

【図３】本発明のアクセス処理の概略のフローチャート

【図４】図３のエラー処理の詳細のフローチャート

【図５】図３のエラー処理の詳細のフローチャート（続き）

【図６】従来装置のブロック図

【符号の説明】

１：上位装置（ホストコンピュータ）

２：ディスクアレイ制御装置

３：上位インタフェース制御部

４：デバイス制御部

５：ディスクアレイ

６：オペレータ制御部

３１：インタフェース制御部

３２，４２：ＭＰＵ

３３，４３：データ転送制御部

３４：カウンタ

３５：フラグレジスタ

３６：不揮発記憶部

３７，４５：マイクロプログラム

３８：上位報告部

３９：ロギング処理部

４１：ディスクアレイ制御部

４４：データチェックカウンタ

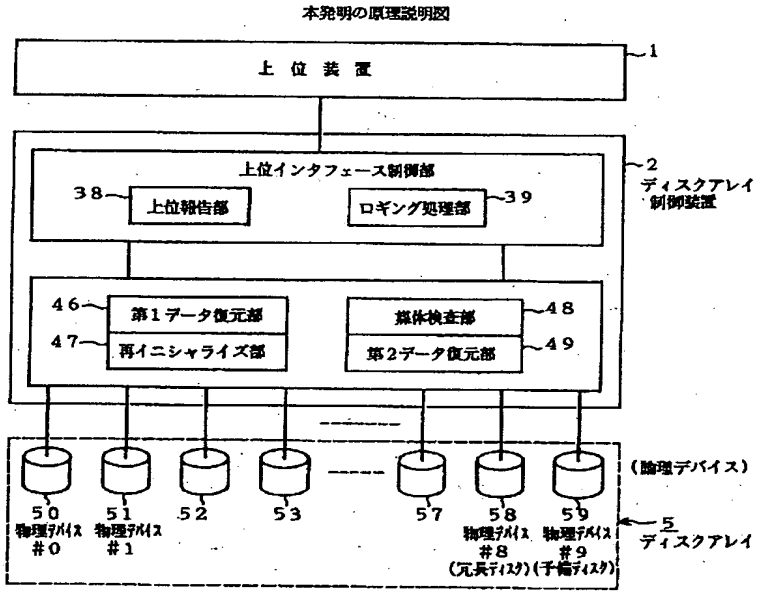
４６：第１データ復元部

４７：再イニシャライズ部

４８：媒体検査部

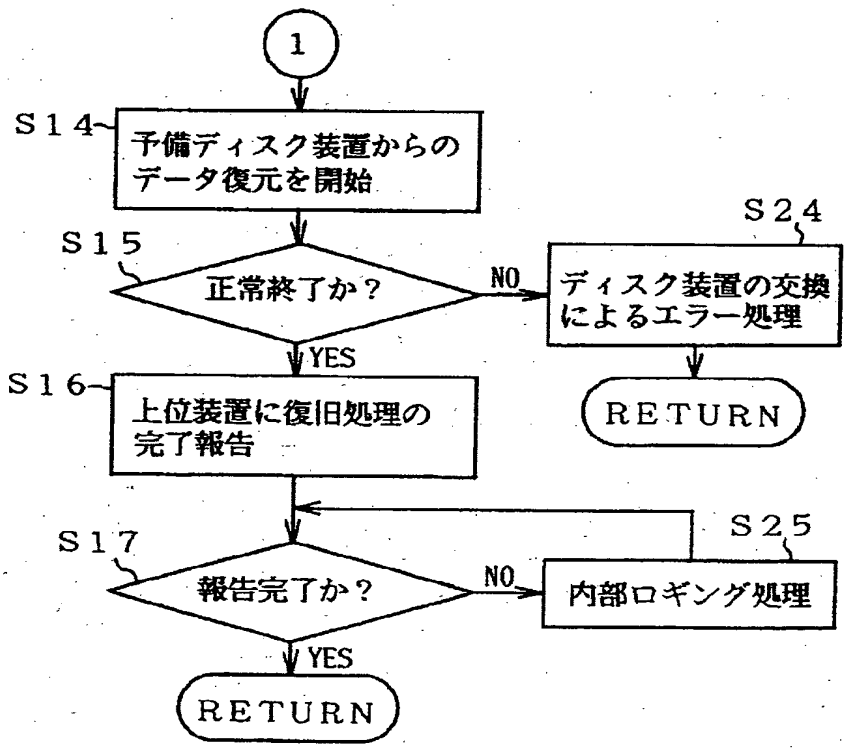
４９：第２データ復元部

【図１】



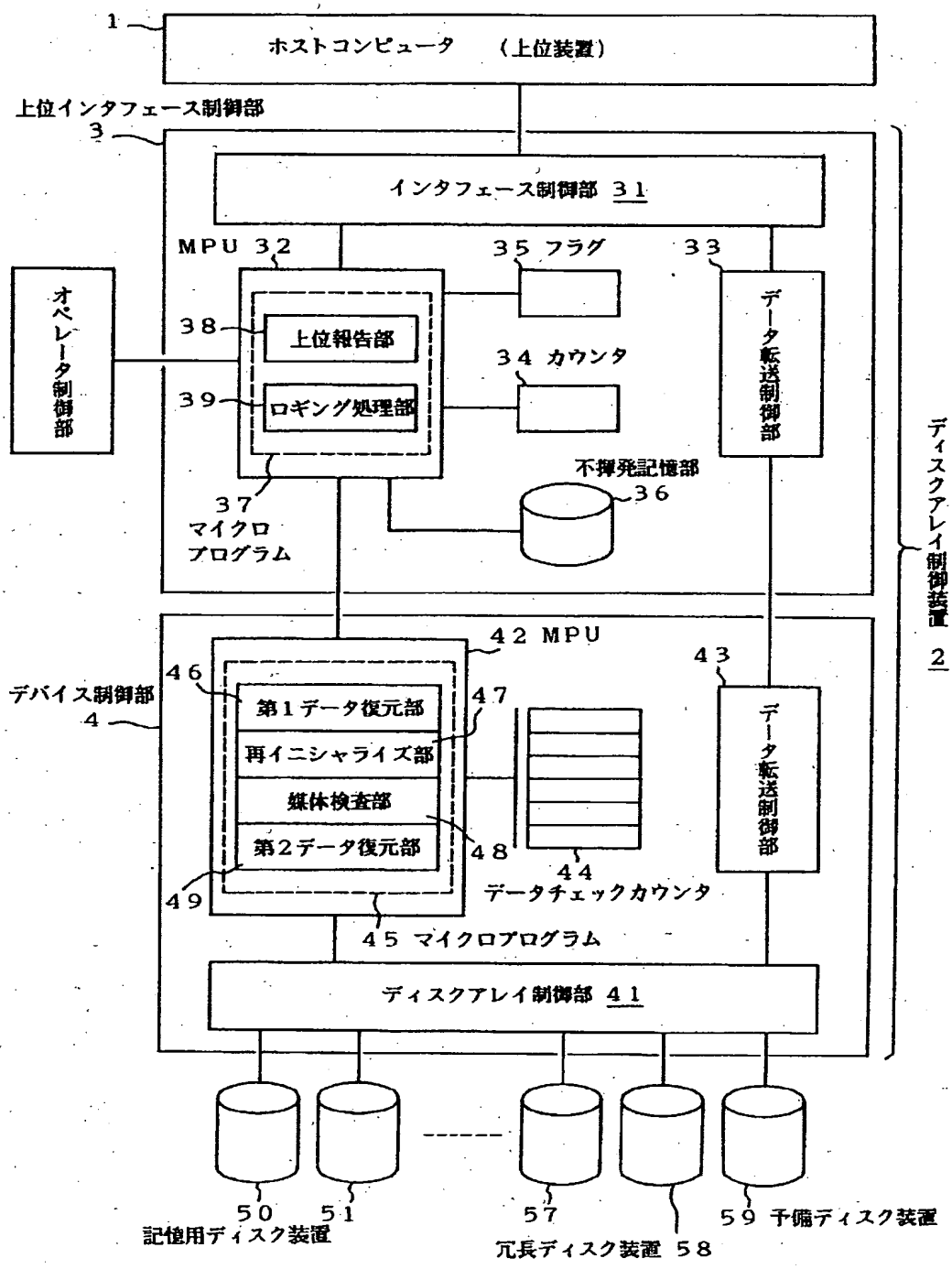
【図５】

図３のエラー処理の詳細のフローチャート（続き）



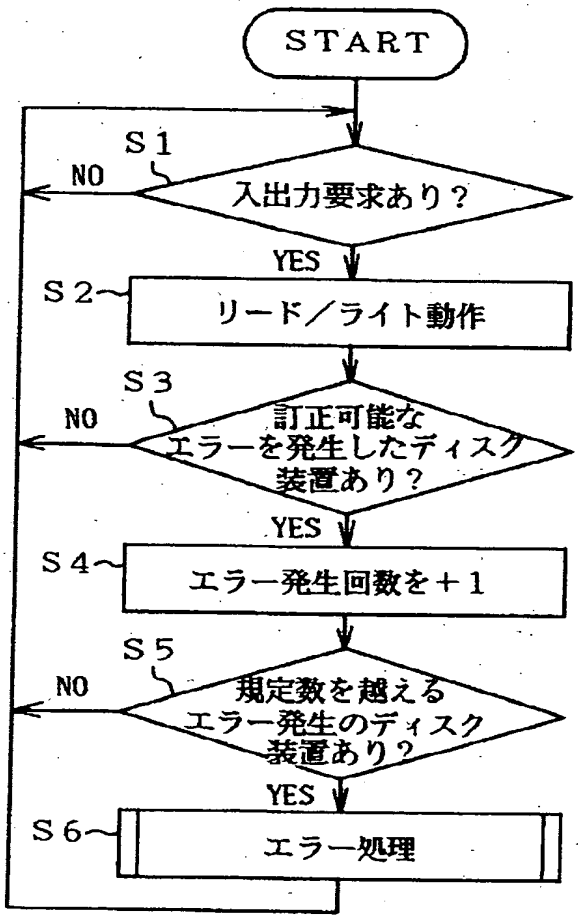
【図２】

本発明の一実施例を示したブロック図

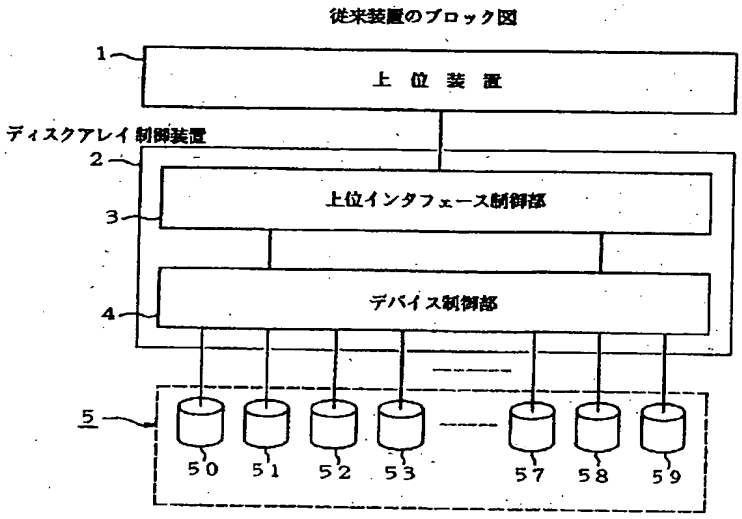


【図３】

本発明のアクセス処理の概略のフローチャート

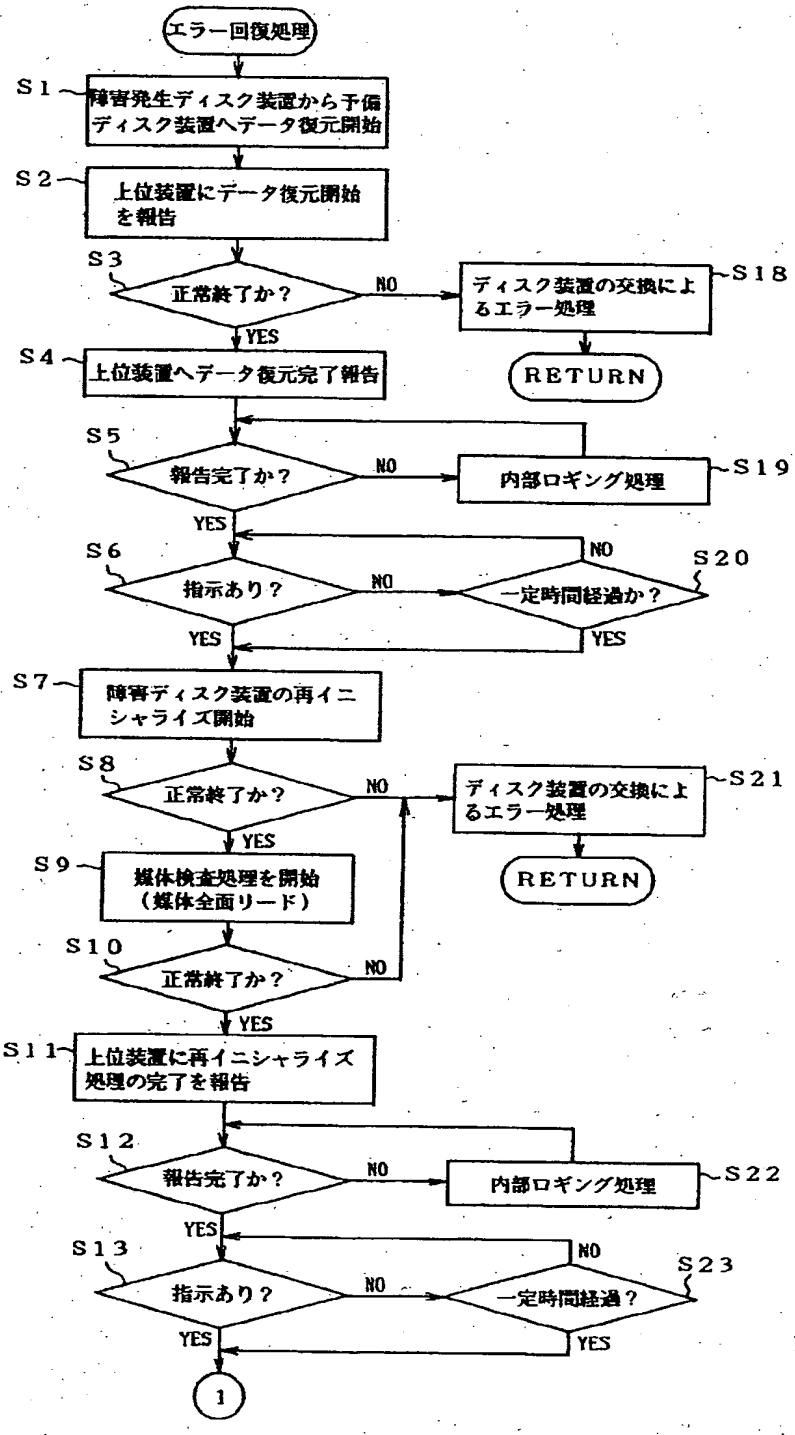


【図６】



【図４】

図３のエラー処理の詳細のフローチャート



A Case for Redundant Arrays of Inexpensive Disks (RAID)

David A. Patterson, Garth Gibson, and Randy H. Katz

Computer Science Division
Department of Electrical Engineering and Computer Sciences
571 Evans Hall
University of California
Berkeley, CA 94720
(pattrsn@ginger.berkeley.edu)

Abstract. Increasing performance of CPUs and memories will be squandered if not matched by a similar performance increase in I/O. While the capacity of Single Large Expensive Disks (SLED) has grown rapidly, the performance improvement of SLED has been modest. Redundant Arrays of Inexpensive Disks (RAID), based on the magnetic disk technology developed for personal computers, offers an attractive alternative to SLED, promising improvements of an order of magnitude in performance, reliability, power consumption, and scalability. This paper introduces five levels of RAIDs, giving their relative cost/performance, and compares RAID to an IBM 3380 and a Fujitsu Super Eagle.

1. Background: Rising CPU and Memory Performance

The users of computers are currently enjoying unprecedented growth in the speed of computers. Gordon Bell said that between 1974 and 1984, single chip computers improved in performance by 40% per year, about twice the rate of minicomputers [Bell 84]. In the following year Bill Joy predicted an even faster growth [Joy 85]:

$$MIPS = 2^{Year-1984}$$

Mainframe and supercomputer manufacturers, having difficulty keeping pace with the rapid growth predicted by "Joy's Law," cope by offering multiprocessors as their top-of-the-line product.

But a fast CPU does not a fast system make. Gene Amdahl related CPU speed to main memory size using this rule [Siewiorek 82]:

Each CPU instruction per second requires one byte of main memory;

If computer system costs are not to be dominated by the cost of memory, then Amdahl's constant suggests that memory chip capacity should grow at the same rate. Gordon Moore predicted that growth rate over 20 years ago:

$$transistors/chip = 2^{Year-1964}$$

As predicted by Moore's Law, RAMs have quadrupled in capacity every two [Moore 75] to three years [Myers 86].

Recently the ratio of megabytes of main memory to MIPS has been defined as *alpha* [Garcia 84], with Amdahl's constant meaning alpha = 1. In part because of the rapid drop of memory prices, main memory sizes have grown faster than CPU speeds and many machines are shipped today with alphas of 3 or higher.

To maintain the balance of costs in computer systems, secondary storage must match the advances in other parts of the system. A key measure of magnetic disk technology is the growth in the maximum number of bits that can be stored per square inch, or the bits per inch in a track times the number of tracks per inch. Called M.A.D., for maximal areal density, the "First Law in Disk Density" predicts [Frank87] :

$$MAD = 10^{(Year-1971)/10}$$

Magnetic disk technology has doubled capacity and halved price every three years, in line with the growth rate of semiconductor memory, and in practice between 1967 and 1979 the disk capacity of the average IBM data processing system more than kept up with its main memory [Stevens81].

Capacity is not the only memory characteristic that must grow rapidly to maintain system balance, since the speed with which instructions and data are delivered to a CPU also determines its ultimate performance. The speed of main memory has kept pace for two reasons:

(1) the invention of caches, showing that a small buffer can be managed automatically to contain a substantial fraction of memory references;
(2) and the SRAM technology, used to build caches, whose speed has improved at the rate of 40% to 100% per year.

In contrast to primary memory technologies, the performance of single large expensive magnetic disks (SLED) has improved at a modest rate. These *mechanical* devices are dominated by the seek and the rotation delays: from 1971 to 1981, the raw seek time for a high-end IBM disk improved by only a factor of two while the rotation time did not change[Harker81]. Greater density means a higher transfer rate when the information is found, and extra heads can reduce the average seek time, but the raw seek time only improved at a rate of 7% per year. There is no reason to expect a faster rate in the near future.

To maintain balance, computer systems have been using even larger main memories or solid state disks to buffer some of the I/O activity. This may be a fine solution for applications whose I/O activity has locality of reference and for which volatility is not an issue, but applications dominated by a high rate of random requests for small pieces of data (such as transaction-processing) or by a low number of requests for massive amounts of data (such as large simulations running on supercomputers) are facing a serious performance limitation.

2. The Pending I/O Crisis

What is the impact of improving the performance of some pieces of a problem while leaving others the same? Amdahl's answer is now known as Amdahl's Law [Amdahl67]:

$$S = \frac{1}{(1-f) + f/k}$$

where:

S = the effective speedup;
f = fraction of work in faster mode; and
k = speedup while in faster mode.

Suppose that some current applications spend 10% of their time in I/O. Then when computers are 10X faster--according to Bill Joy in just over three years--then Amdahl's Law predicts effective speedup will be only 5X. When we have computers 100X faster--via evolution of uniprocessors or by multiprocessors--this application will be less than 10X faster, wasting 90% of the potential speedup.

While we can imagine improvements in software file systems via buffering for near term I/O demands, we need innovation to avoid an I/O crisis [Boral 83].

3. A Solution: Arrays of Inexpensive Disks

Rapid improvements in capacity of large disks have not been the only target of disk designers, since personal computers have created a market for inexpensive magnetic disks. These lower cost disks have lower performance as well as less capacity. Table I below compares the top-of-the-line IBM 3380 model AK4 mainframe disk, Fujitsu M2361A "Super Eagle" minicomputer disk, and the Conner Peripherals CP 3100 personal computer disk.

Characteristics	*IBM 3380*	*Fujitsu M2361A*	*Conners CP3100*	*3380 v. 2361 v 3100 (>1 means 3100 is better)*	
Disk diameter (inches)	14	10.5	3.5	4	3
Formatted Data Capacity (MB)	7500	600	100	.01	.2
Price/MB (controller incl.)	$18-$10	$20-$17	$10-$7	1-2.5	1.7-3
MTTF Rated (hours)	30,000	20,000	30,000	1	1.5
MTTF in practice (hours)	100,000	?	?	?	?
No. Actuators	4	1	1	.2	1
Maximum I/O's/second/Actuator	50	40	30	.6	.8
Typical I/O's/second/Actuator	30	24	20	.7	.8
Maximum I/O's/second/box	200	40	30	.2	.8
Typical I/O's/second/box	120	24	20	.2	.8
Transfer Rate (MB/sec)	3	2.5	1	.3	.4
Power/box (W)	6,600	640	10	660	64
Volume (cu. ft.)	24	3.4	.03	800	110

Table I. *Comparison of IBM 3380 disk model AK4 for mainframe computers, the Fujitsu M2361A "Super Eagle" disk for minicomputers, and the Conners Peripherals CP 3100 disk for personal computers. By "Maximum I/O's/second" we mean the maximum number of average seeks and average rotates for a single sector access. Cost and reliability information on the 3380 comes from widespread experience [IBM 87] [Gawlick87] and the information on the Fujitsu from the manual [Fujitsu 87], while some numbers on the new CP3100 are based on speculation. The price per megabyte is given as a range to allow for different prices for volume discount and different mark-up practices of the vendors. (The 8 watt maximum power of the CP3100 was increased to 10 watts to allow for the inefficiency of an external power supply, since the other drives contain their own power supplies).*

One surprising fact is that the number of I/Os per second per actuator in an inexpensive disk is within a factor of two of the large disks. In several of the remaining metrics, including price per megabyte, the inexpensive disk is superior or equal to the large disks.

The small size and low power are even more impressive since disks such as the CP3100 contain full track buffers *and* most functions of the traditional mainframe controller. Small disk manufacturers can provide such functions in high volume disks because of the efforts of standards committees in defining higher level peripheral interfaces, such as the ANSI X3.131-1986 Small Computer System Interface (SCSI). Such standards have encouraged companies like Adeptec to offer SCSI interfaces as single chips, in turn allowing disk companies to *embed* mainframe controller functions at low cost. Figure 1 compares the traditional mainframe disk approach and the small computer disk approach. The same SCSI interface chip embedded as a controller in every disk can also be used as the direct memory access (DMA) device at the other end of the SCSI bus.

Such characteristics lead to our proposal for building I/O systems as arrays of inexpensive disks, either interleaved for the large transfers of supercomputers [Kim 86][Livny 87][Salem86] or independent for the many small transfers of transaction processing. Using the information in Table I, 75 inexpensive disks potentially have 12 times the I/O bandwidth of the IBM 3380 and the same capacity, with lower power consumption and cost.

4. Caveats

We cannot explore all issues associated with such arrays in the space available for this paper, so we concentrate on fundamental estimates of price-performance and reliability. Our reasoning is that if there are no advantages in price-performance or terrible disadvantages in reliability, then there is no need to explore further. We characterize a transaction-processing workload to evaluate performance of a collection of inexpensive disks, but remember that such a collection is just one hardware component of a complete tranaction-processing system. While designing a complete TPS based on these ideas is enticing, we will resist that temptation in this paper. Cabling and packaging, certainly an issue in the cost and reliability of an array of many inexpensive disks, is also beyond this paper's scope.

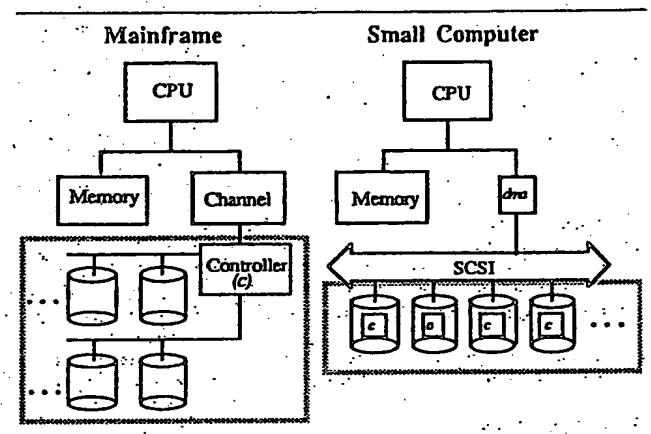


Figure 1. *Comparison of organizations for typical mainframe and small computer disk interfaces. Single chip SCSI interfaces such as the Adaptec AIC-6250 allow the small computer to use a single chip to be the DMA interface as well as provide an embedded controller for each disk [Adeptec 87]. (The price per megabyte in Table I includes everything in the shaded boxes above.)*

5. And Now The Bad News: Reliability

The unreliability of disks forces computer systems managers to make backup versions of information quite frequently in case of failure. What would be the impact on reliability of having a hundredfold increase in disks? Assuming a constant failure rate--that is, an exponentially distributed time to failure--and that failures are independent--both assumptions made by disk manufacturers when calculating the Mean Time To Failure (MTTF)--the reliability of an array of disks is:

$$\text{MTTF of a Disk Array} = \frac{\text{MTTF of a Single Disk}}{\text{Number of Disks in the Array}}$$

Using the information in Table I, the MTTF of 100 CP 3100 disks is 30,000/100 = 300 hours, or less than 2 weeks. Compared to the 30,000 hour (> 3 years) MTTF of the IBM 3380, this is dismal. If we consider scaling the array to 1000 disks, then the MTTF is 30 hours or about one day, requiring an adjective worse than dismal.

Without fault tolerance, large arrays of inexpensive disks are too unreliable to be useful.

6. A Better Solution: RAID

To overcome the reliability challenge, we must make use of extra disks containing redundant information to recover the original information when a disk fails. Our acronym for these Redundant Arrays of Inexpensive Disks is *RAID*. To simplify the explanation of our final proposal and to avoid confusion with previous work, we give a taxonomy of five different organizations of disk arrays, beginning with mirrored disks and progressing through a variety of alternatives with differing performance and reliability. We refer to each organization as a RAID *level*.

The reader should be forewarned that we describe all levels as if implemented in hardware solely to simplify the presentation, for RAID ideas are applicable to software implementations as well as hardware.

Reliability. Our basic approach will be to break the arrays into reliability groups, with each group having extra "check" disks containing redundant information. When a disk fails we assume that within a short time the failed disk will be replaced and the information will be

reconstructed on to the new disk using the redundant information. This time is called the mean time to repair (MTTR). The MTTR can be reduced if the system includes extra disks to act as "hot" standby spares; when a disk fails, a replacement disk is switched in electronically. Periodically a human operator replaces all failed disks. Here are other terms that we use:

D = total number of disks with data (not including extra check disks);
G = number of data disks in a *group* (not including extra check disks);
C = number of check disks in a group;
$n_G = D/G$ = number of groups;

As mentioned above we make the same assumptions that disk manufacturers make--that failures are exponential and independent. (An earthquake or power surge is a situation where an array of disks might not fail independently.) Since these reliability predictions will be very high, we want to emphasize that the reliability is only of the the disk-head assemblies with this failure model, and not the whole software and electronic system. In addition, in our view the pace of technology means extremely high MTTF are "overkill"--for, independent of expected lifetime, users will replace obsolete disks. After all, how many people are still using 20 year old disks?

The general MTTF calculation for single-error repairing RAID is given in two steps. First, the group MTTF is:

$$MTTF_{Group} = \frac{MTTF_{Disk}}{G+C} * \frac{1}{\text{Probability of another failure in a group before repairing the dead disk}}$$

As more formally derived in the appendix, the probability of a second failure before the first has been repaired is:

$$\text{Probability of Another Failure} = \frac{MTTR}{MTTF_{Disk}/(\text{No. Disks}-1)} = \frac{MTTR}{MTTF_{Disk}/(G+C-1)}$$

The intuition behind the formal calculation in the appendix comes from trying to calculate the average number of second disk failures during the repair time for X single disk failures. Since we assume that disk failures occur at a uniform rate, this average number of second failures during the repair time for X first failures is

$$\frac{X * MTTR}{\text{MTTF of remaining disks in the group}}$$

The average number of second failures for a single disk is then

$$\frac{MTTR}{MTTF_{Disk}/\text{No. of remaining disks in the group}}$$

The MTTF of the remaining disks is just the MTTF of a single disk divided by the number of good disks in the group, giving the result above.

The second step is the reliability of the whole system, which is approximately (since $MTTF_{Group}$ is not quite distributed exponentially):

$$MTTF_{RAID} = \frac{MTTF_{Group}}{n_G}$$

Plugging it all together, we get:

$$MTTF_{RAID} = \frac{MTTF_{Disk}}{G+C} * \frac{MTTF_{Disk}}{(G+C-1)*MTTR} * \frac{1}{n_G}$$

$$= \frac{(MTTF_{Disk})^2}{(G+C)*n_G * (G+C-1)*MTTR}$$

$$MTTF_{RAID} = \frac{(MTTF_{Disk})^2}{(D+C*n_G)*(G+C-1)*MTTR}$$

Since the formula is the same for each level, we make the abstract numbers concrete using these parameters as appropriate: D=100 total data disks, G=10 data disks per group, $MTTF_{Disk}$ = 30,000 hours, $MTTR$ = 1 hour, with the check disks per group C determined by the RAID level.

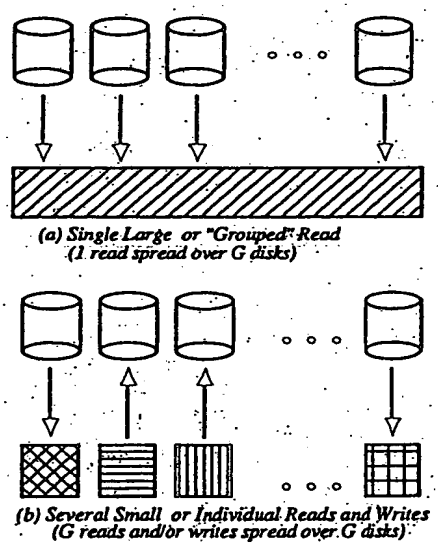
Reliability Overhead Cost. This is simply the extra check disks, expressed as a percentage of the number of data disks D. As we shall see below, the cost varies with RAID level from 100% down to 4%.

Useable Storage Capacity Percentage. Another way to express this reliability overhead is in terms of the percentage of the total capacity of data disks *and* check disks that can be used to store data. Depending on the organization, this varies from a low of 50% to a high of 96%.

Performance. Since supercomputer applications and transaction-processing systems have different access patterns and rates, we need different metrics to evaluate both. For supercomputers we count the number of reads and writes per second for large blocks of data, with large defined as getting at least one sector from each data disk in a group. During large transfers all the disks in a group act as a single unit, each reading or writing a portion of the large data block in parallel.

A better measure for transaction-processing systems is the number of individual reads or writes per second. Since transaction-processing systems (e.g., debits/credits) use a read-modify-write sequence of disk accesses, we include that metric as well. Ideally during small transfers each disk in a group can act independently, either reading or writing independent information. In summary supercomputer applications need a *high data rate* while transaction-processing need a *high I/O rate.*

For both the large and small transfer calculations we assume the minimum user request is a sector, that a sector is small relative to a track, and that there is enough work to keep every device busy. Thus sector size affects both disk storage efficiency and transfer size. Figure 2 shows the ideal operation of large and small disk accesses in a RAID.



(a) Single Large or "Grouped" Read (1 read spread over G disks)

(b) Several Small or Individual Reads and Writes (G reads and/or writes spread over G disks)

Figure 2. *Large transfer vs. small transfers in a group of G disks.*

The six performance metrics are then the number of reads, writes, and read-modify-writes per second for both large (grouped) or small (individual) transfers. Rather than give absolute numbers for each metric, we calculate efficiency: the number of events per second for a RAID relative to the corresponding events per second for a single disk. (This is Boral's I/O bandwidth per gigabyte [Boral 83] scaled to gigabytes per disk.) In this paper we are after fundamental differences so we use simple, deterministic throughput measures for our performance metric rather than latency.

Effective Performance Per Disk. The cost of disks can be a large portion of the cost of a database system, so the I/O performance per disk--factoring in the overhead of the check disks--suggests the cost/performance of a system. This is the bottom line for a RAID.

7. First Level RAID: Mirrored Disks

Mirrored disks are a traditional approach for improving reliability of magnetic disks. This is the most expensive option we consider since all disks are duplicated (G=1 and C=1), and every write to a data disk is also a write to a check disk. Tandem doubles the number of controllers for fault tolerance, allowing an optimized version of mirrored disks that lets reads occur in parallel. Table II shows the metrics for a Level 1 RAID assuming this optimization.

MTTF	Exceeds Useful Product Lifetime (4,500,000 hrs or > 500 years)	
Total Number of Disks	2D	
Overhead Cost	100%	
Useable Storage Capacity	50%	
Events/Sec vs. Single Disk	*Full RAID*	*Efficiency Per Disk*
Large (or Grouped) Reads	2D/S	1.00/S
Large (or Grouped) Writes	D/S	.50/S
Large (or Grouped) R-M-W	4D/3S	.67/S
Small (or Individual) Reads	2D	1.00
Small (or Individual) Writes	D	.50
Small (or Individual) R-M-W	4D/3	.67

Table II. *Characteristics of Level 1 RAID. Here we assume that writes are not slowed by waiting for the second write to complete because the slowdown for writing 2 disks is minor compared to the slowdown S for writing a whole group of 10 to 25 disks. Unlike a "pure" mirrored scheme with extra disks that are invisible to the software, we assume an optimized scheme with twice as many controllers allowing parallel reads to all disks, giving full disk bandwidth for large reads and allowing the reads of read-modify-writes to occur in parallel.*

When individual accesses are distributed across multiple disks, average queueing, seek, and rotate delays may differ from the single disk case. Although bandwidth may be unchanged, it is distributed more evenly, reducing variance in queueing delay and, if the disk load is not too high, also reducing the expected queueing delay through parallelism [Livny 87]. When many arms seek to the same track then rotate to the described sector, the average seek and rotate time will be larger than the average for a single disk, tending toward the worst case times. This affect should not generally more than double the average access time to a single sector while still getting many sectors in parallel. In the special case of mirrored disks with sufficient controllers, the choice between arms that can read any data sector will reduce the time for the average read seek by up to 45% [Bitton 88].

To allow for these factors but to retain our fundamental emphasis we apply a slowdown factor, S, when there are more than two disks in a group. In general, $1 \le S \le 2$ whenever groups of disk work in parallel. With synchronous disks the spindles of all disks in the group are synchronous so that the corresponding sectors of a group of disks pass under the heads simultaneously,[Kurzweil 88] so for synchronous disks there is no slowdown and $S = 1$. Since a Level 1 RAID has only one data disk in its group, we assume that the large transfer requires the same number of disks acting in concert as found in groups of the higher level RAIDs: 10 to 25 disks.

Duplicating all disks can mean doubling the cost of the database system or using only 50% of the disk storage capacity. Such largess inspires the next levels of RAID.

8. Second Level RAID: Hamming Code for ECC

The history of main memory organizations suggests a way to reduce the cost of reliability. With the introduction of 4K and 16K DRAMs, computer designers discovered that these new devices were subject to losing information due to alpha particles. Since there were many single bit DRAMs in a system and since they were usually accessed in groups of 16 to 64 chips at a time, system designers added redundant chips to correct single errors and to detect double errors in each group. This increased the number of memory chips by 12% to 38%--depending on the size of the group--but it significantly improved reliability.

As long as all the data bits in a group are read or written together, there is no impact on performance. However, reads of less than the group size require reading the whole group to be sure the information is correct, and writes to a portion of the group mean three steps:

1) a read step to get all the rest of the data;
2) a modify step to merge the new and old information;
3) a write step to write the full group, including check information.

Since we have scores of disks in a RAID and since some accesses are to groups of disks, we can mimic the DRAM solution by bit-interleaving the data across the disks of a group and then add enough check disks to detect and correct a single error. A single parity disk can detect a single error, but to correct an error we need enough check disks to identify the disk with the error. For a group size of 10 data disks (G) we need 4 check disks (C) in total, and if G = 25 then C = 5 [Hamming50]. To keep down the cost of redundancy, we assume the group size will vary from 10 to 25.

Since our individual data transfer unit is just a sector, bit-interleaved disks mean that a large transfer for this RAID must be at least G sectors. Like DRAMs, reads to a smaller amount implies reading a full sector from each of the bit-interleaved disks in a group, and writes of a single unit involve the read-modify-write cycle to all the disks. Table III shows the metrics of this Level 2 RAID.

MTTF		Exceeds Useful Lifetime			
		G=10 (494,500 hrs or >50 years)		*G=25* (103,500 hrs or 12 years)	
Total Number of Disks		1.40D		1.20D	
Overhead Cost		40%		20%	
Useable Storage Capacity		71%		83%	
Events/Sec	*Full RAID*	*Efficiency Per Disk*		*Efficiency Per Disk*	
(vs. Single Disk)		*L2*	*L2/L1*	*L2*	*L2/L1*
Large Reads	*D/S*	.71/S	71%	.86/S	86%
Large Writes	*D/S*	.71/S	143%	.86/S	172%
Large R-M-W	*D/S*	.71/S	107%	.86/S	129%
Small Reads	*D/SG*	.07/S	6%	.03/S	3%
Small Writes	*D/2SG*	.04/S	6%	.02/S	3%
Small R-M-W	*D/SG*	.07/S	9%	.03/S	4%

Table III. *Characteristics of a Level 2 RAID. The L2/L1 column gives the % performance of level 2 in terms of level 1 (>100% means L2 is faster). As long as the transfer unit is large enough to spread over all the data disks of a group, the large I/Os get the full bandwidth of each disk, divided by S to allow all disks in a group to complete. Level 1 large reads are faster because data is duplicated and so the redundancy disks can also do independent accesses. Small I/Os still require accessing all the disks in a group, so only D/G small I/Os can happen at a time, again divided by S to allow a group of disks to finish. Small Level 2 writes are like small R-M-W because full sectors must be read before new data can be written onto part of each sector.*

For large writes, the level 2 system has the same performance as level 1 even though it uses fewer check disks, and so on a per disk basis it outperforms level 1. For small data transfers the performance is dismal either for the whole system or per disk; all the disks of a group must be accessed for a small transfer, limiting the maximum number of simultaneous accesses to D/G. We also include the slowdown factor S since the access must wait for all the disks to complete.

Thus level 2 RAID is desirable for supercomputers but inappropriate for transaction processing systems, with increasing group size increasing the disparity in performance per disk for the two applications. In recognition of this fact, Thinking Machines Incorporated announced a Level 2 RAID this year for its Connection Machine supercomputer called the "Data Vault," with G = 32 and C = 8, including one hot standby spare [Hillis 87].

Before improving small data transfers, we concentrate once more on lowering the cost.

9. Third Level RAID: Single Check Disk Per Group

Most check disks in the level 2 RAID are used to determine which disk failed, for only one redundant parity disk is needed to detect an error. These extra disks are truly "redundant" since most disk controllers can already detect if a disk failed: either through special signals provided in the disk interface or the extra checking information at the end of a sector used to detect and correct soft errors. So information on the failed disk can be reconstructed by calculating the parity of the remaining good disks and then comparing bit-by-bit to the parity calculated for the original full

group. When these two parities agree, the failed bit was a 0; otherwise it was a 1. If the check disk is the failure, just read all the data disks and store the group parity in the replacement disk.

Reducing the check disks to one per group (C=1) reduces the overhead cost to between 4% and 10% for the group sizes considered here. The performance for the third level RAID system is the same as the Level 2 RAID, but the effective performance per disk increases since it needs fewer check disks. This reduction in total disks also increases reliability, but since it is still larger than the useful lifetime of disks, this is a minor point. One advantage of a level 2 system over level 3 is that the extra check information associated with each sector to correct soft errors is not needed, increasing the capacity per disk by perhaps 10%. Level 2 also allows all soft errors to be corrected "on the fly" without having to reread a sector. Table IV summarizes the third level RAID characteristics and Figure 3 compares the sector layout and check disks for levels 2 and 3.

MTTF		Exceeds Useful Lifetime			
		G=10 (820,000 hrs or >90 years)		*G=25* (346,000 hrs. or 40 years)	
Total Number of Disks		1.10D		1.04D	
Overhead Cost		10%		4%	
Useable Storage Capacity		91%		96%	
Events/Sec (vs. Single Disk)	*Full RAID*	*Efficiency Per Disk L3*	*L3/L2*	*L3/L1*	*Efficiency Per Disk L3* *L3/L2* *L3/L1*
Large Reads	*D/S*	.91/S	127%	91%	.96/S 112% 96%
Large Writes	*D/S*	.91/S	127%	182%	.96/S 112% 192%
Large R-M-W	*D/S*	.91/S	127%	136%	.96/S 112% 142%
Small Reads	*D/SG*	.09/S	127%	8%	.04/S 112% 3%
Small Writes	*D/2SG*	.05/S	127%	8%	.02/S 112% 3%
Small R-M-W	*D/SG*	.09/S	127%	11%	.04/S 112% 5%

Table IV. *Characteristics of a Level 3 RAID. The L3/L2 column gives the % performance of L3 in terms of L2 and the L3/L1 column gives it in terms of L1 (>100% means L3 is faster). The performance for the full systems is the same in RAID levels 2 and 3, but since there are fewer check disks the performance per disk improves.*

Park and Balasubramanian proposed a third level RAID system without suggesting a particular application [Park86]. Our calculations suggest it is a much better match to supercomputer applications than to transaction processing systems. This year two disk manufacturers have announced level 3 RAIDs for such applications using synchronized 5.25 inch disks with G=4 and C=1: one from Maxtor and one from Micropolis [Maginnis 87].

This third level has brought the reliability overhead cost to its lowest level, so in the last two levels we improve performance of small accesses without changing cost or reliability.

10. Fourth Level RAID: Independent Reads/Writes

Spreading a transfer across all disks within the group has the following advantage:

- Large or grouped transfer time is reduced because transfer bandwidth of the entire array can be exploited.

But it has the following disadvantages as well:

- Reading/writing to a disk in a group requires reading/writing to all the disks in a group; levels 2 and 3 RAIDs can perform only one I/O at a time per group.
- If the disks are not synchronized, you do not see average seek and rotational delays; the observed delays should move towards the worst case, hence the *S* factor in the equations above.

This fourth level RAID improves performance of small transfers through parallelism--the ability to do more than one I/O per group at a time. We no longer spread the individual transfer information across several disks, but keep each individual unit in a single disk.

The virtue of bit-interleaving is the easy calculation of the Hamming code needed to detect or correct errors in level 2. But recall that in the third level RAID we rely on the disk controller to detect errors within a single disk sector. Hence, if we store an individual transfer unit in a single sector, we can detect errors on an individual read without accessing any other disk. Figure 3 shows the different ways the information is stored in a sector for RAID levels 2, 3, and 4. By storing a whole transfer unit in a sector, reads can be independent and operate at the maximum rate of a disk yet still detect errors. Thus the primary change between level 3 and 4 is that we interleave data between disks at the sector level rather than at the bit level.

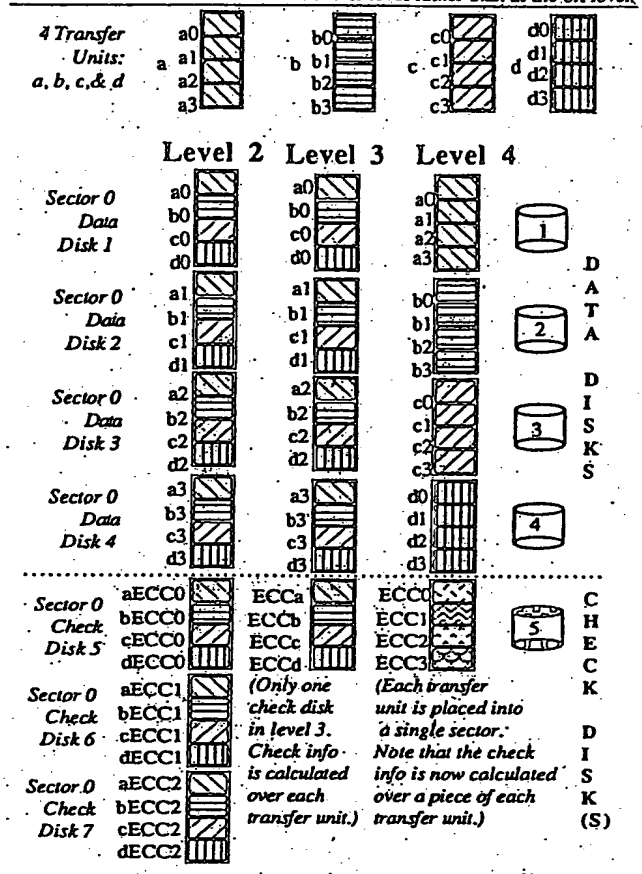


Figure. 3. *Comparison of location of data and check information in sectors for RAID levels 2, 3, and 4 for G=4. Not shown is the small amount of check information per sector added by the disk controller to detect and correct soft errors within a sector. Remember that we use physical sector numbers and hardware control to explain these ideas, but RAID can be implemented by software using logical sectors and disks.*

At first thought you might expect that an individual write to a single sector still involves all the disks in a group since (1) the check disk must be rewritten with the new parity data, and (2) the rest of the data disks must be read to be able to calculate the new parity data. Recall that each parity bit is just a single exclusive OR of all the corresponding data bits in a group. In level 4 RAID, unlike level 3, the parity calculation is much simpler since, if we know the old data value and the old parity value as well as the new data value, we can calculate the new parity information as follows:

new parity = (old data xor new data) xor old parity

In level 4 a small write then uses 2 disks to perform 4 accesses--2 reads and 2 writes--while a small read involves only one read on one disk. Table V summarizes the fourth level RAID characteristics. Note that all small accesses improve--dramatically for the reads--but the small read-modify-write is still so slow relative to a level 1 RAID that its applicability to transaction processing is doubtful. Recently Salem and Garcia-Molina proposed a Level 4 system [Salem 86].

Before proceeding to the next level we need to explain the performance of small writes in Table V (and hence small read-modify-writes since they entail the same operations in this RAID). The formula for the small writes divides *D* by 2 instead of 4 because 2

accesses can proceed in parallel: the old data and old parity can be read at the same time and the new data and new parity can be written at the same time. The performance of small writes is also divided by G because the single check disk in a group must be read and written with every small write in that group, thereby limiting the number of writes that can be performed at a time to the number of groups.

The check disk is the bottleneck, and the final level RAID removes this bottleneck.

MTTF	Exceeds Useful Lifetime	
	G=10 (820,000 hrs or >90 years)	*G=25* (346,000 hrs or 40 years)
Total Number of Disks	1.10D	1.04D
Overhead Cost	10%	4%
Useable Storage Capacity	91%	96%

Events/Sec (vs. Single Disk)	*Full RAID*	*Efficiency Per Disk L4*	*L4/L3*	*L4/L1*	*Efficiency Per Disk L4*	*L4/L3*	*L4/L1*
Large Reads	*D/S*	.91/S	100%	91%	.96/S	100%	96%
Large Writes	*D/S*	.91/S	100%	182%	.96/S	100%	192%
Large R-M-W	*D/S*	.91/S	100%	136%	.96/S	100%	146%
Small Reads	*D*	.91	1200%	91%	.96	3000%	96%
Small Writes	*D/2G*	.05	120%	9%	.02	120%	4%
Small R-M-W	*D/G*	.09	120%	14%	.04	120%	6%

Table V. *Characteristics of a Level 4 RAID. The L4/L3 column gives the % performance of L4 in terms of L3 and the L4/L1 column gives it in terms of L1 (>100% means L4 is faster). Small reads improve because they no longer tie up a whole group at a time. Small writes and R-M-Ws improve some because we make the same assumptions as we made in Table II: the slowdown for two related I/Os can be ignored because only two disks are involved.*

11. Fifth Level RAID: No Single Check Disk

While level 4 RAID achieved parallelism for reads, writes are still limited to one per group since every write must read and write the check disk. The final level RAID distributes the data and check information across all the disks--including the check disks. Figure 4 compares the location of check information in the sectors of disks for levels 4 and 5 RAIDs.

The performance impact of this small change is large since RAID level 5 can support multiple individual writes per group. For example, suppose in Figure 4 above we want to write sector 0 of disk 2 and sector 1 of disk 3. As shown on the left Figure 4, in RAID level 4 these writes must be sequential since both sector 0 and sector 1 of disk 5 must be written. However, as shown on the right, in RAID level 5 the writes can proceed in parallel since a write to sector 0 of disk 2 still involves a write to disk 5 but a write to sector 1 of disk 3 involves a write to disk 4.

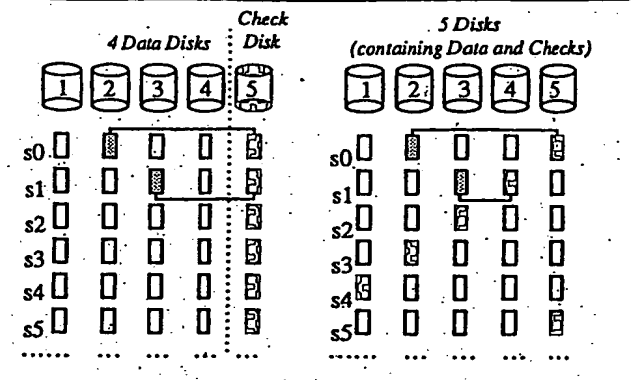
These changes bring RAID level 5 near the best of both worlds: small read-modify-writes now perform close to the speed per disk of a level 1 RAID while keeping the large transfer performance per disk and high useful storage capacity percentage of the RAID levels 3 and 4. Spreading the data across all disks even improves the performance of small reads, since there is one more disk per group that contains data. Table VI summarizes the characteristics of this RAID.

Keeping in mind the caveats given earlier, a Level 5 RAID appears very attractive if you want to do just supercomputer applications, or just transaction processing when storage capacity is limited, or if you want to do both supercomputer applications *and* transaction processing.

12. Discussion

Before concluding the paper, we wish to note a few more interesting points about RAIDs. The first is that while the schemes for disk striping and parity support were presented as if they were done by hardware, there is no necessity to do so. We just give the method, and the decision between hardware and software solutions is strictly one of cost and benefit. For example, in cases where disk buffering is effective, there is no extra disks reads for level 5 small writes since the old data and old parity would be in main memory, so software would give the best performance as well as the least cost.

In this paper we have assumed the transfer unit is a multiple of the sector. As the size of the smallest transfer unit grows larger than one



(a) Check information for Level 4 RAID for G=4 and C=1. The sectors are shown below the disks. (The checked areas indicate the check information.) Writes to s0 of disk 2 and s1 of disk 3 imply writes to s0 and s1 of disk 5. The check disk (5) becomes the write bottleneck.

(b) Check information for Level 5 RAID for G=4 and C=1. The sectors are shown below the disks, with the check information and data spread evenly through all the disks. Writes to s0 of disk 2 and s1 of disk 3 still imply 2 writes, but they can be split across 2 disks: to s0 of disk 5 and to s1 of disk 4.

Figure 4. *Location of check information per sector for Level 4 RAID vs. Level 5 RAID.*

MTTF	Exceeds Useful Lifetime	
	G=10 (820,000 hrs or >90 years)	*G=25* (346,000 hrs or 40 years)
Total Number of Disks	1.10D	1.04D
Overhead Cost	10%	4%
Useable Storage Capacity	91%	96%

Events/Sec (vs. Single Disk)	*Full RAID*	*Efficiency Per Disk L5*	*L5/L4*	*L5/L1*	*Efficiency Per Disk L5*	*L5/L4*	*L5/L1*
Large Reads	*D/S*	.91/S	100%	91%	.96/S	100%	96%
Large Writes	*D/S*	.91/S	100%	182%	.96/S	100%	192%
Large R-M-W	*D/S*	.91/S	100%	136%	.96/S	100%	144%
Small Reads	*(1+C/G)D*	1.00	110%	100%	1.00	104%	100%
Small Writes	*(1+C/G)D/4*	.25	550%	50%	.25	1300%	50%
Small R-M-W	*(1+C/G)D/2*	.50	550%	75%	.50	1300%	75%

Table VI. *Characteristics of a Level 5 RAID. The L5/L4 column gives the % performance of L5 in terms of L4 and the L5/L1 column gives it in terms of L1 (>100% means L5 is faster). Because reads can be spread over all disks, including what were check disks in level 4, all small I/Os improve by a factor of 1+C/G. Small writes and R-M-Ws improve because they are no longer constrained by group size, getting the full disk bandwidth for the 4 I/O's associated with these accesses. We again make the same assumptions as we made in Tables II and V: the slowdown for two related I/Os can be ignored because only two disks are involved.*

sector per drive--such as a full track with an I/O protocol that supports data returned out-of-order--then the performance of RAIDs improves significantly because of the full track buffer in every disk. For example, if every disk begins transferring to its buffer as soon as it reaches the next sector, then S may reduce to less than 1 since there would be virtually no rotational delay. With transfer units the size of a track, it is not even clear if synchronizing the disks in a group improves RAID performance.

This paper makes two separable points: the advantages of building I/O systems from personal computer disks and the advantages of five different disk array organizations, independent of disks used in those array. The later point starts with the traditional mirrored disks to achieve acceptable reliability, with each succeeding level improving:

- *the data rate*, characterized by a small number of requests per second for massive amounts of sequential information (supercomputer applications);

- *the I/O rate*, characterized by a large number of read-modify-writes to a small amount of random information (transaction-processing);
- or *the useable storage capacity*;

or possibly all three.

Figure 5 shows the performance improvements per disk for each level RAID. The highest performance per disk comes from either Level 1 or Level 5. In transaction-processing situations using no more than 50% of storage capacity, then the choice is mirrored disks (Level 1). However, if the situation calls for using more than 50% of storage capacity, or for supercomputer applications, or for combined supercomputer applications and transaction processing, then Level 5 looks best. Both the strength and weakness of Level 1 is that it duplicates data rather than calculating check information, for the duplicated data improves read performance but lowers capacity and write performance, while check data is useful only on a failure.

Inspired by the space-time product of paging studies [Denning 78], we propose a single figure of merit called the *space-speed product*: the useable storage fraction times the efficiency per event. Using this metric, Level 5 has an advantage over Level 1 of 1.7 for reads and 3.3 for writes for G=10.

Let us return to the first point, the advantages of building I/O system from personal computer disks. Compared to traditional Single Large Expensive Disks (SLED), Redundant Arrays of Inexpensive Disks (RAID) offer significant advantages for the same cost. Table VII compares a level 5 RAID using 100 inexpensive data disks with a group size of 10 to the IBM 3380. As you can see, a level 5 RAID offers a factor of roughly 10 improvement in performance, reliability, and power consumption (and hence air conditioning costs) and a factor of 3 reduction in size over this SLED. Table VII also compares a level 5 RAID using 10 inexpensive data disks with a group size of 10 to a Fujitsu M2361A "Super Eagle". In this comparison RAID offers roughly a factor of 5 improvement in performance, power consumption, and size with more than two orders of magnitude improvement in (calculated) reliability.

RAID offers the further advantage of modular growth over SLED. Rather than being limited to 7,500 MB per increase for $100,000 as in the case of this model of IBM disk, RAIDs can grow at either the group size (1000 MB for $11,000) or, if partial groups are allowed, at the disk size (100 MB for $1,100). The flip side of the coin is that RAID also makes sense in systems considerably smaller than a SLED. Small incremental costs also makes hot standby spares practical to further reduce MTTR and thereby increase the MTTF of a large system. For example, a 1000 disk level 5 RAID with a group size of 10 and a few standby spares could have a calculated MTTF of over 45 years.

A final comment concerns the prospect of designing a complete transaction processing system from either a Level 1 or Level 5 RAID. The drastically lower power per megabyte of inexpensive disks allows systems designers to consider battery backup for the whole disk array--the power needed for 110 PC disks is less than two Fujitsu Super Eagles. Another approach would be to use a few such disks to save the contents of battery backed-up main memory in the event of an extended power failure. The smaller capacity of these disks also ties up less of the database during reconstruction, leading to higher availability. (Note that Level 5 ties up all the disks in a group in event of failure while Level 1 only needs the single mirrored disk during reconstruction, giving Level 1 the edge in availability).

13. Conclusion

RAIDs offer a cost effective option to meet the challenge of exponential growth in the processor and memory speeds. We believe the size reduction of personal computer disks is a key to the success of disk arrays, just as Gordon Bell argues that the size reduction of microprocessors is a key to the success in multiprocessors [Bell 85]. In both cases the smaller size simplifies the interconnection of the many components as well as packaging and cabling. While large arrays of mainframe processors (or SLEDs) are possible, it is certainly easier to construct an array from the same number of microprocessors (or PC drives). Just as Bell coined the term "multi" to distinguish a multiprocessor made from microprocessors, we use the term "RAID" to identify a disk array made from personal computer disks.

With advantages in cost-performance, reliability, power consumption, and modular growth, we expect RAIDs to replace SLEDs in future I/O systems. There are, however, several open issues that may bare on the practicality of RAIDs:

- *What is the impact of a RAID on latency?*
- *What is the impact on MTTF calculations of non-exponential failure assumptions for individual disks?*
- *What will be the real lifetime of a RAID vs. calculated MTTF using the independent failure model?*
- *How would synchronized disks affect level 4 and 5 RAID performance?*
- *How does "slowdown" S actually behave? [Livny 87]*
- *How do defective sectors affect RAID?*
- *How do you schedule I/O to level 5 RAIDs to maximize write parallelism?*
- *Is there locality of reference of disk accesses in transaction processing?*
- *Can information be automatically redistributed over 100 to 1000 disks to reduce contention?*
- *Will disk controller design limit RAID performance?*
- *How should 100 to 1000 disks be constructed and physically connected to the processor?*
- *What is the impact of cabling on cost, performance, and reliability?*
- *Where should a RAID be connected to a CPU so as not to limit performance? Memory bus? I/O bus? Cache?*
- *Can a file system allow differ striping policies for different files?*
- *What is the role of solid state disks and WORMs in a RAID?*
- *What is the impact on RAID of "parallel access" disks (access to every surface under the read/write head in parallel)?*

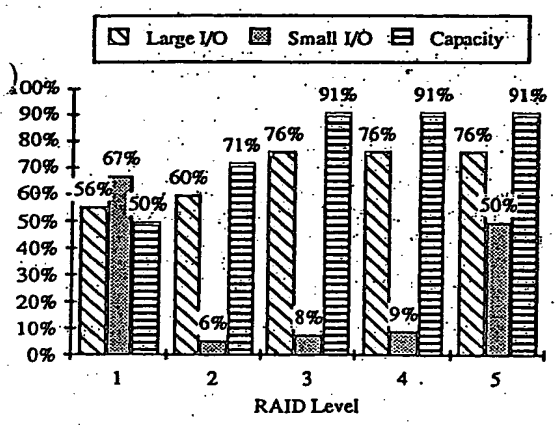


Figure 5. *Plot of Large (Grouped) and Small (Individual) Read-Modify-Writes per second per disk and useable storage capacity for all five levels of RAID (D=100, G=10). We assume a single S factor uniformly for all levels, with S=1.3 where it is needed.*

Characteristics	*RAID 5L (100,10) (CP3100)*	*SLED (IBM 3380)*	*RAID v. SLED (>1 better for RAID)*	*RAID 5L (10,10) (CP3100)*	*SLED (Fujitsu M2361)*	*RAID v. SLED (>1 better for RAID)*
Formatted Data Capacity (MB)	10,000	7,500	1.33	1,000	600	1.67
Price/MB (controller incl.)	$11-$8	$18-$10	2.2-.9	$11-$8	$20-$17	2.5-1.5
Rated MTTF (hours)	820,000	30,000	27.3	8,200,000	20,000	410
MTTF in practice (hours)	?	100,000	?	?	?	?
No. Actuators	110	4	22.5	11	1	11
Max I/O's/Actuator	30	50	.6	30	40	.8
Max Grouped RMW/box	1250	100	12.5	125	20	6.2
Max Individual RMW/box	825	100	8.2	83	20	4.2
Typ I/O's/Actuator	20	30	.7	20	24	.8
Typ Grouped RMW/box	833	60	13.9	83	12	6.9
Typ Individual RMW/box	550	60	9.2	55	12	4.6
Volume/Box (cubic feet)	10	24	2.4	1	3.4	3.4
Power/box (W)	1100	6,600	6.0	110	640	5.8
Min. Expansion Size (MB)	100-1000	7,500	7.5-75	100-1000	600	0.6-6

Table VII. *Comparison of IBM 3380 disk model AK4 to Level 5 RAID using 100 Conners & Associates CP 3100s disks and a group size of 10 and a comparison of the Fujitsu M2361A "Super Eagle" to a level 5 RAID using 10 inexpensive data disks with a group size of 10. Numbers greater than 1 in the comparison columns favor the RAID.*

Acknowledgements

We wish to acknowledge the following people who participated in the discussions from which these ideas emerged: Michael Stonebraker, John Ousterhout, Doug Johnson, Ken Lutz, Anapum Bhide, Gaetano Boriello, Mark Hill, David Wood, and students in SPATS seminar offered at U. C. Berkeley in Fall 1987. We also wish to thank the following people who gave comments useful in the preparation of this paper: Anapum Bhide, Pete Chen, Ron David, Dave Ditzel, Fred Douglis, Dieter Gawlick, Jim Gray, Mark Hill, Doug Johnson, Joan Pendleton, Martin Schulze, and Hervé Touati. This work was supported by the National Science Foundation under grant # MIP-8715235.

Appendix: Reliability Calculation

Using probability theory we can calculate the $MTTF_{Group}$. We first assume independent and exponential failure rates. Our model uses a biased coin with the probability of heads being the probability that a second failure will occur within the MTTR of a first failure. Since disk failures are exponential:

$$\text{Probability(at least one of the remaining disks failing in MTTR)} = [\,1 - (e^{-MTTR/MTTF_{Disk}})^{(G+C-1)}\,]$$

In all practical cases

$$MTTR << \frac{MTTF_{Disk}}{G+C}$$

and since $(1 - e^{-X})$ is approximately X for $0 < X << 1$:

$$\text{Probability(at least one of the remaining disks failing in MTTR)} = MTTR*(G+C-1)/MTTF_{Disk}$$

Then that on a disk failure we flip this coin:

heads => a system crash, because a second failure occurs before the first was repaired;

tails => recover from error and continue.

Then

$$MTTF_{Group} = \text{Expected[Time between Failures]} * \text{Expected[no. of flips until first heads]}$$

$$= \frac{\text{Expected[Time between Failures]}}{\text{Probability(heads)}}$$

$$= \frac{MTTF_{Disk}}{(G+C)*(MTTR*(G+C-1)/MTTF_{Disk})}$$

$$MTTF_{Group} = \frac{(MTTF_{Disk})^2}{(G+C)*(G+C-1)*MTTR}$$

Group failure is not precisely exponential in our model, but we have validated this simplifying assumption for practical cases of MTTR << MTTF/(G+C). This makes the MTTF of the whole system just $MTTF_{Group}$ divided by the number of groups, n_G.

References

[Bell 84] C.G. Bell, "The Mini and Micro Industries," *IEEE Computer*, Vol. 17, No. 10 (October 1984), pp. 14-30.

[Joy 85] B. Joy, presentation at ISSCC '85 panel session, Feb. 1985.

[Siewiorek 82] D.P. Siewiorek, C.G. Bell, and A. Newell, *Computer Structures: Principles and Examples*, p. 46.

[Moore 75] G.E. Moore, "Progress in Digital Integrated Electronics," *Proc. IEEE Digital Integrated Electronic Device Meeting*, (1975), p. 11.

[Myers 86] G.J. Myers, A.Y.C. Yu, and D.L. House, "Microprocessor Technology Trends," *Proc. IEEE*, Vol. 74, no. 12, (December 1986), pp. 1605-1622.

[Garcia 84] H. Garcia-Molina, R. Cullingford, P. Honeyman, R. Lipton, "The Case for Massive Memory," Technical Report 326, Dept. of EE and CS, Princeton Univ., May 1984.

[Myers 86] W. Myers, "The Competitiveness of the United States Disk Industry," *IEEE Computer*, Vol. 19, No. 11 (January 1986), pp. 85-90.

[Frank 87] P.D. Frank, "Advances in Head Technology," presentation at *Challenges in Disk Technology Short Course*, Institute for Information Storage Technology, Santa Clara University, Santa Clara, California, December 15-17, 1987.

[Stevens 81] L.D. Stevens, "The Evolution of Magnetic Storage," *IBM Journal of Research and Development*, Vol. 25, No. 5, Sept. 1981, pp. 663-675.

[Harker 81] J.M. Harker *et al.*, "A Quarter Century of Disk File Innovation," *ibid.*, pp. 677-689.

[Amdahl 67] G.M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," *Proceedings AFIPS 1967 Spring Joint Computer Conference* Vol. 30 (Atlantic City, New Jersey April 1967), pp. 483-485.

[Boral 83] H. Boral and D.J. DeWitt, "Database Machines: An Ideas Whose Time Has Passed? A Critique of the Future of Database Machines," *Proc. International Conf. on Database Machines*, Edited by H.-O. Leilich and M. Misskoff, Springer-Verlag, Berlin, 1983.

[IBM 87] "IBM 3380 Direct Access Storage Introduction," IBM GC 26-4491-0, September 1987.

[Gawlick 87] D. Gawlick, private communication, Nov., 1987.

[Fujitsu 87] "M2361A Mini-Disk Drive Engineering Specifications," (revised) Feb., 1987, B03P-4825-0001A.

[Adaptec 87] AIC-6250, *IC Product Guide*, Adaptec, stock # DB0003-00 rev. B, 1987, p. 46.

[Livny 87] Livny, M., S. Khoshafian, H. Boral, "Multi-disk management algorithms," *Proc. of ACM SIGMETRICS*, May 1987.

[Kim 86] M.Y. Kim, "Synchronized disk interleaving," *IEEE Trans. on Computers*, vol. C-35, no. 11, Nov. 1986.

[Salem 86] K. Salem and Garcia-Molina, H., "Disk Striping," *IEEE 1986 Int. Conf. on Data Engineering*, 1986.

[Bitton 88] D. Bitton and J. Gray, "Disk Shadowing," *in press*, 1988.

[Kurzweil 88] F. Kurzweil, "Small Disk Arrays - The Emerging Approach to High Performance," presentation at Spring COMPCON 88, March 1, 1988, San Francisco, CA.

[Hamming 50] R. W. Hamming, "Error Detecting and Correcting Codes," *The Bell System Technical Journal*, Vol XXVI, No. 2 (April 1950), pp. 147-160.

[Hillis 87] D. Hillis, private communication, October, 1987.

[Park 86] A. Park and K. Balasubramanian, "Providing Fault Tolerance in Parallel Secondary Storage Systems," Department of Computer Science, Princeton University, CS-TR-057-86, Nov. 7, 1986.

[Maginnis 87] N.B. Maginnis, "Store More, Spend Less: Mid-range Options Abound," *Computerworld*, Nov. 16, 1987, p. 71.

[Denning 78] P.J. Denning and D.R. Slutz, "Generalized Working Sets for Segment Reference Strings," *CACM*, vol. 21, no. 9, (Sept. 1978) pp. 750-759.

[Bell 85] Bell, C.G., "Multis: a new class of multiprocessor computers," *Science*, vol. 228 (April 26, 1985) 462-467.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☑ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.